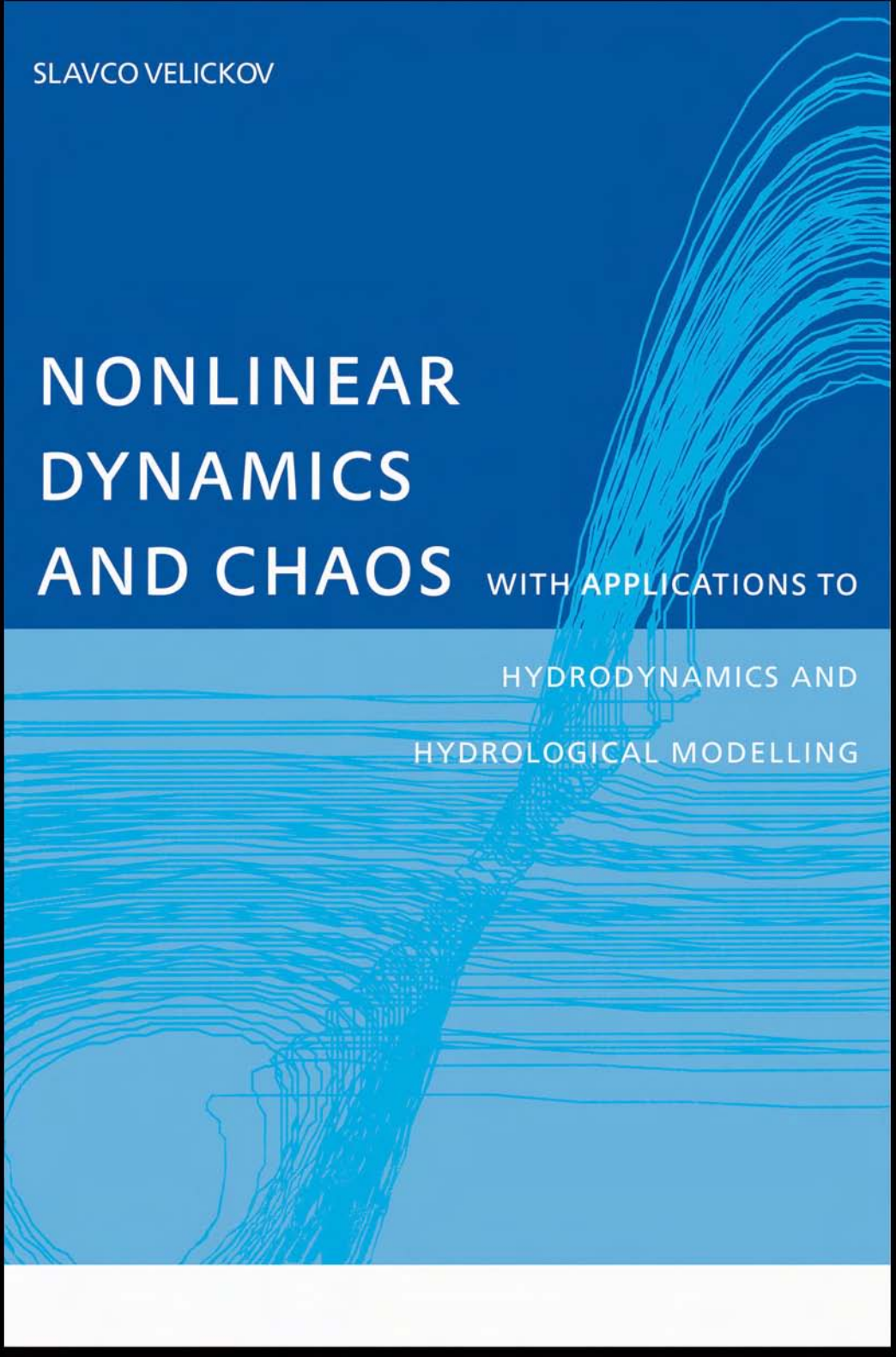


# NONLINEAR DYNAMICS AND CHAOS WITH APPLICATIONS TO HYDRODYNAMICS AND HYDROLOGICAL MODELLING

# Nonlinear Dynamics and Chaos with Applications to Hydrodynamics and Hydrological Modelling

DISSERTATION
Submitted in fulfilment of the requirements of
the Board for the Doctorate of Delft University of Technology
and the Academic Board of the UNESCO-IHE Institute for Water Education
for the Degree of DOCTOR
to be defended in public
on Tuesday, 18 May at 10:30 hours
in Delft, The Netherlands
*By*

*SLAVCO VELICKOV*
*born in Stip, Macedonia*
*BSc. in Civil Engineering (University of St. Ciril & Methody, Macedonia)*
*MSc. in Hydraulic Engineering (University of St. Ciril & Methody, Macedonia)*
*MSc. in Hydroinformatics (IHE Delft, The Netherlands)*

This dissertation has been approved by the promoter
Prof. dr. R.K.Price TU Delft/UNESCO-IHE Delft, The Netherlands
Members of the Awarding Committee:
Chairman Rector Magnificus TU Delft, The Netherlands
Co-chairman Rector UNESCO-IHE Delft, The Netherlands
Prof. dr. ir. M.Stive TU Delft, The Netherlands
Prof. dr. ir. R.Cooke TU Delft, The Netherlands
Dr. ir. D.P.Solomatine UNESCO-IHE Delft, The Netherlands
Prof. dr. D.P.Loucks Cornell University, USA
Prof. dr. A.M.Mynnet UNESCO-IHE Delft, The Netherlands



Published by A.A.Balkema Publishers, a member of Taylor & Francis Group plc. www.balkema.nl and www.tandf.co.uk
This edition published in the Taylor & Francis e-Library, 2006.



ISBN 0-203-02432-X Master e-book ISBN

ISBN 90 5809 691 2 (Print Edition) (Taylor & Francis Group)

# Contents

# Abstract

A hydroinformatics system represents an electronic knowledge encapsulator that models part of the real world and can be used for the simulation and analysis of physical, chemical and biological processes in water systems, for a better management of the aquatic environment. Thus, modelling is at the heart of hydroinformatics. The theory of nonlinear dynamics and chaos and the extent to which recent improvements in the understanding of inherently nonlinear natural processes present challenges to the use of mathematical models in the analysis of water and environmental systems are elaborated in this work. In particular, it demonstrates that the deterministic chaos present in many nonlinear systems can impose fundamental limitations on our ability to predict behaviour even when well-defined mathematical models exist. On the other hand, methodologies and tools from the theory of nonlinear dynamics and chaos can provide means for a better accuracy of short-term predictions as demonstrated through the practical applications in this work.

The first chapter discusses the role of mathematical modelling in hydroinformatics, exemplifying both the physically-based and the data-driven modelling practices and challenges. It further elaborates the main goal of this research work in describing, elaborating mathematically and illustrating the general principles and concepts of modelling based on chaos theory. It also addresses the implications that arise from modelling complex nonlinear dynamical systems in the aquatic environment that are essential for understanding the possible consequences of nonlinearity for modelling.

Modelling nonlinear dynamical systems based on chaos theory is closely connected to data-driven modelling. Chapter 2 describes the history of learning models from data and critically reviews both the classical approaches based on empirical risk minimisation and the approaches based on structural risk minimisation. Learning models from data as an illposed problem that is closely related to computational intelligence based on search and optimisation methods. These are discussed further in this chapter.

Chapter 3 is at the heart of this work. It describes, elaborates mathematically and illustrates the main concepts of the theory of nonlinear dynamics and deterministic chaos. It further introduces and demonstrates the methods and techniques for the identification, reconstruction, delineation and quantification of the underlying dynamics of nonlinear dynamical systems from a time series of observables. The phase-space reconstruction based on univariate time series is further extended and elaborated using the multivariate embedding methodology proposed in this work. Finally, it elaborates how models can be constructed that realistically map the underlying structure dictating the dynamical evolution of the system.

Chapter 4 further extends this notion of models that learn from data by introducing the Bayesian network formalism. Special attention is given to dynamic Bayesian networks that are well suited for learning models from time series data observed on complex dynamical systems.

In Chapter 5, a novel hybrid framework for modelling nonlinear dynamical systems that draws on both chaos theory and dynamic Bayesian networks is proposed, mathematically elaborated and demonstrated. This modelling framework combines the multivariate phase-space reconstruction of the underlying dynamics based on a time series of observables and a mixture of local models learned in a dynamic Bayesian network formalism.

In Chapter 6, the proposed modelling framework is applied to the identification, modelling and prediction of hydrodynamical and hydrological systems: sea water level and surge dynamics along the Dutch cost, precipitation dynamics at De Bilt meteorological station in the Netherlands and rainfall-runoff dynamics of the Huai river in China. The results from these applications show that the methodology and the modelling framework presented in this thesis generate reliable and accurate short-term forecasts and can be used as a valuable modelling tool in engineering practice.

# Samenvatting

Een hydroinformatica-systeem is een weergave van een elektronisch kennisraamwerk waarmee een gedeelte van de werkelijkheid wordt gemodelleerd en dat gebruikt kan worden voor de simulatie en analyse van fysische, chemische en biologische processen in water systemen ten behoeve van een beter beheer van de aquatische omgeving

Aldus vormt modellering het centrum van de hydroinformatica. Dit proefschrift behandelt de theorie van de niet-lineaire dynamica en chaostheorie. De aan deze methoden onlosmakelijk verbonden niet-lineaire natuurlijke processen, vormen in het bijzonder een uitdaging voor het toepassen van mathematische modellen voor de analyse van nietlineaire processen in land- en watersystemen.

In het bijzonder wordt gedemonstreerd dat deterministische chaos, die aanwezig is in vele niet-lineaire systemen, fundamentele beperkingen kan opleggen aan ons vermogen om gedrag te voorspellen, zelfs als er sprake is van goed gedefinieerde mathematische modellen. Daarentegen laten de praktische toepassingen die in dit proefschrift zijn beschreven zien dat methodieken en technieken uit de theorie van de niet-lineaire dynamica en chaos, een basis kunnen vormen voor grotere nauwkeurigheid van korte termijn voorspellingen.

Het eerste hoofdstuk beschrijft de rol van mathematische modellering in de hydroinformatica, waarbij voorbeelden worden gegeven van zowel op fysica gebaseerde (physically-based) als op gegevens gebaseerde (data-driven) modeltoepassingen en de daarbij horende uitdagingen. Verder wordt het hoofddoel van het onderzoek geformuleerd, ondersteund met beschrijving, mathematische formulering, en voorbeelden van de algemene principes en concepten van het modelleren op basis van chaostheorie. Tevens worden de implicaties behandeld van het modelleren van complexe niet-lineaire dynamische systemen in de aquatische omgeving, die essentieel zijn voor het begrijpen van de mogelijke consequenties van niet-lineariteit bij het modelleren.

Het modelleren van niet-lineaire dynamische systemen, gebaseerd op chaostheorie, sluit nauw aan op gegevens-gestuurd modelleren. Hoofdstuk 2 beschrijft de historie van dergelijke modellen en behandelt op kritische wijze de beschikbare literatuur van zowel de klassieke benaderingen die zijn gebaseerd op empirische risico minimalisatie als de benaderingen die zijn gebaseerd op structurele risico minimalisatie. Gegevens-gestuurd modelleren is een niet scherp gedefinieerd probleemgebied dat nauw gerelateerd is aan kunstmatige intelligentie, welke is gebaseerd op zoekalgoritmen en optimalisatiemethoden. Dit wordt verder in dit hoofdstuk beschreven.

Hoofdstuk 3 behandelt het centrale deel van het onderzoek. Het geeft een mathematische beschrijving van de hoofdprincipes van de theorie van de niet-lineaire dynamica en deterministische chaos. Voorts worden methoden en technieken geïntroduceerd voor de identificatie, reconstructie, beschrijving en kwantificering van de onderliggende structuur van niet-lineaire dynamische systemen.

De fase-ruimte reconstructie die is gebaseerd op unvariabele tijdreeks benadering, is in dit onderzoek verder uitgewerkt tot een multivariabele methodologie. Tenslotte wordt in

dit hoofdstuk behandeld hoe lokale modellen kunnen worden opgebouwd die op realistische wijze de onderliggende fase-ruimte structuur weergeven die de dynamische evolutie van het systeem bepaalt.

In hoofdstuk 4 wordt de notie van modellen die leren uit gegevens, verder uitgewerkt aan de hand van de introductie van Bayesiaanse netwerken. Speciale aandacht wordt geschonken aan dynamische Bayesiaanse netwerken die zeer geschikt zijn voor het gegevens-gestuurd leren uit tijdreeksen die zijn ontleend aan complexe dynamische systemen.

In hoofdstuk 5 wordt een nieuw hybride raamwerk voor het modelleren van niet-lineaire dynamische systemen voorgesteld, mathematisch uitgewerkt en beproefd, dat bouwt op zowel chaostheorie als op dynamische Bayesiaanse netwerken. Dit modelleerraamwerk combineert de multivariabele fase-ruimte reconstructie van de onderliggende dynamica, die is gebaseerd op een tijdreeks van waarnemingen, met een mix van lokale modellen in een dynamisch Bayesiaans netwerk.

In hoofdstuk 6 wordt het voorgestelde modelleerraamwerk toegepast voor identificatie, modellering en voorspelling van hydrodynamische en hydrologische systemen: voorspelling van zeewaterstand en golfbeweging langs de Nederlandse kust; neerslagdynamica van het meteorologische station De Bilt; en regen-afvoer dynamica van de Huai rivier in China. Het resultaat van deze toepassingen toont aan dat de methodiek en het in dit proefschrift voorgestelde modelleerraamwerk betrouwbare en nauwkeurige kortetermijn voorspellingen genereert. Het ontwikkelde modelleerraamwerk kan worden beschouwd als waardevol instrumentarium voor de ingenieurspraktijk.

# Acknowledgements

It is almost 11 years since Prof. Z.Skoklevski of the Faculty of Civil Engineering in Skopje, Macedonia, encouraged and supported me to continue my education at the International Institute for Infrastructural Hydraulic and Environmental Engineering, from whom I have privileged to enjoy tremendous support for my study and further career development in the last 6 years. Sadly, Prof. Skoklevski was not able to see this thesis.

I am indebted to my promoter Prof. Roland Price for creating the research opportunities and for his always invaluable and inspiring discussions and conversations throughout these years.

I would like to thank Dr Dimitri Solomatine for his continuous supervision and support, Ir. Jan Luijendijk, Prof. Arthur Mynett, Prof. Michael Abbott and Dr. Arnold Lobbrecht, for the valuable discussions at various stages and not only related to this research work. I thank all professors from the Examination Committee for reading and improving the final manuscript.

I wish further to thank all of my colleagues and friends in the hydroinformatics core and IHE, who in different ways supported and encouraged me. Special thanks to my, first of all, friend and colleague Dr. Andreja Jonoski for his constant support.

Finally, to my family, I wish to express my gratefulness for their continuous love and understanding during all these years.

# Chapter 1
# Introduction

*The scientist does not study nature because it is useful; he studies it because he delights in it, and he delights in it because it is beautiful. If nature were not beautiful, it would not be worth knowing, and if nature were not worth knowing, life would not be worth living. Of course I do not here speak of that beauty that strikes the senses, the beauty of qualities and appearances; not that I undervalue such beauty, far from it, but it has nothing to do with science; I mean that profounder beauty which comes from the harmonious order of the parts, and which human intelligence can grasp.*

—Henri Poincaré

## 1.1 Modelling: the current practices and challenges

In a thesis like "nonlinear dynamics and deterministic chaos and its applications to hydrodynamics and hydrological modelling" it is natural to ask what contribution one of the fundamental technologies of modern science—namely, *mathematical modelling,* (typically including extensive numerical simulations)—can bring to efforts made to enhance our understanding of natural processes and phenomena in the aquatic environment. A moment's consideration makes it clear that the potential contribution is profound: for instance, one can immediately identify a number of water and environmental problems—the motion of the water in the oceans, the generation and mitigation of floods, sediment transport and morphodynamics, water quality etc.—of overwhelming importance in which an accurate quantitative description of the causal relationships between specific processes, actions and consequences can only be obtained from studies of highly sophisticated mathematical models containing many subtle and interacting effects.

These mathematical models are mainly conceptualisations of the primary *physical processes* that are perceived and identified to be deterministic in their contribution to the natural phenomenon, expressed through mathematical algorithmic equations. Such equations describe the quantitative relationships between the different system parameters, and thus the behaviour of the whole system, based on fundamental principles, such as conservation of mass, momentum (and energy). The solution of these equations, in order to find the functional relationships that describe and define the physical boundary domain in which the water flows, requires the application of specific numerical techniques and the imposition of certain boundary conditions. This branch of science that considers the *discretisation* of the physical domain and the corresponding equations governing the

natural processes, was conceived after the Second World War and born in the 1960s, and further known as *computational hydraulics* (or computational fluid dynamics) is now well established.

By bringing these computational hydraulics techniques together with the recently proliferating information and communication technologies, a new discipline emerged of what is nowadays referred to as *Hydroinformatics* (Abbott, 1991). A hydroinformatics system indicates, as Abbott put it, an *electronic knowledge encapsulator* that models part of the real world and can be used for the simulation and analysis of physical, chemical and biological processes in water, for a better management of the aquatic environment. Therefore, the development of mathematical models, which adequately represent our current image of reality, is at the heart of hydroinformatics (Price, 2001).

But hydroinformatics is even more complex, in that it is an emerging *socio-technical construct* (see Jonoski, 2002). This leads to the modelling of socio-political issues—for example, socio-economic consequences of certain activities in the aquatic environment or the involvement of different stakeholders and public participation in the decision making processes in the management of limited water resources. In addition to the complex technical issues one must also try to account for the vagaries of human psychology. We can expect that issues based on global water-related problems, as well as the general consequences of limited resources and credible, degrading environmental conditions, will become increasingly relevant factors. These have to be taken into account in the future development of hydroinformatics systems.

Each of these issues involves many individual components and processes, interacting with each other in complex ways. Clearly one immediate, primarily technical, challenge to mathematical modelling is to quantify these interactions. For instance, in water quality modelling, the challenge is to extend the forms of the algorithmic equations used to conceptualise the processes of advection and dispersion to include sediment, chemical and biological processes, which are less well understood than the water hydrodynamics alone. Such technical questions will—and should—remain the purview of experts, and in most cases their resolution requires the successful collaboration of experts from many different disciplines. This leads to the convergence of different sciences and the notion of *integrated modelling*. In this area, the challenge is to develop well-defined computational models, properly reflecting the essential governing processes and features of such complex problems.

Beyond this problem-specific technical challenge, however, are challenges and *limitations* that arise from the very nature of dynamical systems in which many elements, some of which may adapt their behaviour in time, are interacting. Looking back at the organisation of the classical sciences, we find that at each level of understanding, basically we study two types of phenomena: (i) *agents* (molecules, finite volumes, cells, species and recently software modules) and (ii) *interaction of agents* (chemical reactions, physical interactions and processes, system responses, emergence and evolution). Studying agents in isolation is a fruitful way of discovering insights into the form, function and conceptualisation of an agent, but doing so has also some limitations. Although *reductionism* is a powerful way of looking into the natural processes and phenomena, specifically reductionism fails when we try to use it in a reverse direction. As we shall see throughout this thesis, having a complete and perfect understanding of the dynamics of an agent in no way guarantees that we will be able to predict how this

single agent will behave for all the time in future, and especially in the context of other interacting agents. Such adaptive complex dynamical systems often behave in ways that seem non-intuitive, or even counter-intuitive, based on our current knowledge and experience. The reason for this, is of course, real limitations to the extent with which computational models may be applied. For example, the derivation of the original hydrodynamic equations has to make certain assumptions due to our limited knowledge of the underlying processes, such as the resistance and turbulence in particular. Such assumptions are usually expressed in empirical forms that require the values of one or more parameters to be identified in each particular application during the "calibration" process. This requires the results of the computational models to agree closely with observed data. It is important that this calibration process does not violate the physical integrity of the parameters. The procedure of forcing the model parameters in order to reproduce the observed data is due to the fact that the mathematical model is just an *approximate* conceptualisation and representation of the real world systems. The model errors include missing processes and parameters and/or limited knowledge about representations and governing laws about the processes, the error of discretisation of both the physical domain and equations, arbitrary numerical processes depending on the applied numerical scheme, bugs in the numerical code, errors in the measured data, and so on. Thus, there is a need for the modeller to acknowledge and cope with *uncertainties*. Figure 1.1 schematically illustrates how the science expressed through mathematical modelling interacts with the real world.

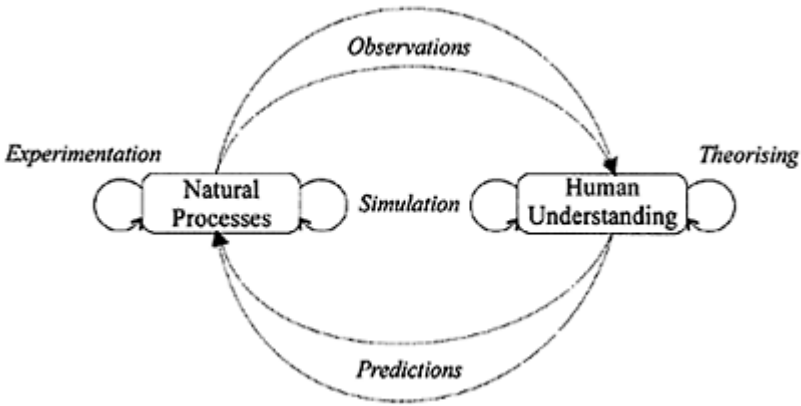



Figure 1.1. The universe of natural processes related to the scientific understanding expressed through the mathematical modelling.

On the left side of the figure are the natural processes that are recurrently coupled to themselves. On the right side is the human understanding that attempts to model the natural world. Experimentation (conducted in laboratory or scale models) consists of manipulating the environment and observing the changes. Furthermore it also implicitly

includes monitoring and collecting data about the real world processes that are measurable and of interest. Theorising is the process of constructing and manipulating models based on the application of (physical) laws about the perceived and identified underlying processes with an ultimate goal of making accurate predictions of future observations. Simulation, mostly done with computational models, somewhat resides between the two, and manipulates both models and environment trying to answer "what-if" questions and scenarios.

The inherently nonlinear nature of these natural processes means that they can exhibit sudden and dramatic changes in the form of their behaviour when small changes are made to the parameters describing the interactions within the system and/or in their initial (boundary) conditions. Further *emergent properties*—that is, characteristics whose existence is not at all apparent in the initial formulation of the system, frequently arise, and theories of *self-organisation* in natural systems (Haken 1983) have attempted to analyse certain aspects of this behaviour. While non-expert users of these computational models can hardly expect—nor be expected—to be aware of the subtle details surrounding the technical modelling aspects, it is vital that those responsible for making decisions on possible courses of action in the aquatic environment be aware in particular of this second category of general constraints and characteristics that affect the *applicability* and *reliability* of the models.

To achieve this awareness, it is essential to go beyond our conventional *linear* intuition and to develop an appreciation of what can—as well as what cannot—occur in complex adaptive nonlinear systems. The development of the appropriate *nonlinear intuition* is extremely important, for it is clear that mathematical models, to the extent that they are credible, not only tell us what is likely to occur but can limit our perceptions of what *can* actually occur. Indeed, in our later discussion of the history of classical modelling based on the Newtonian mechanics, we will exemplify this (potentially negative) aspect of modelling. The essence of this phenomenon is that even in dynamical systems whose evolution from moment to moment follows precise deterministic laws, with no external random influences of any kind, the behaviour over long times can be essentially unpredictable and irregular. That a system governed by deterministic laws can exhibit effectively random-like behaviour runs directly counter to our normal intuition! Perhaps it is because this intuition is inherently *linear;* indeed, this phenomenon cannot occur for linear systems. Linear methods interpret all regular structures in a data set, such as dominant frequency, as linear relationships. This means that the intrinsic dynamics is of the system are govern by the linear paradigm that small causes lead to small effects. Since linear equations describing dynamical system can only lead to exponentially growing or periodically oscillating solutions (dynamical evolution of the system), all irregular behaviour of the system has to be attributed to some random external input to the system. On the other hand, as we will demonstrate throughout this work, random input is not the only possible source of irregularity in a system's output: nonlinear dynamical systems can produce very irregular data with purely deterministic equations of motions, caused by slight changes in some of the control parameters and the sensitivity to the initial (and/or boundary) conditions. Of course, the systems which exhibits both, nonlinearity and random input, will most likely produce irregular data as well.

The ‘small causes-small effects’ intuition is, more likely, because of our view of a clockwork universe, a view which in the past was vigorously stated by the great French mathematician and natural philosopher Laplace; in *Philosophical Essays on Probabilities*, Laplace wrote:

> “An intellect which at any given moment knew all the forces that animate Nature and the mutual positions of the beings that comprise it, if this intellect were vast enough to submit its data to analysis, could condense into a single formula the movement of the greatest bodies of the universe and that of the lightest atom; for such an intellect, nothing could be uncertain; and the future just like the past would be present before its eyes.”

In short, Laplace argued that from a knowledge of the initial state of the universe (and its forces) comes an exact knowledge of the final state of the universe. Indeed, in Newtonian mechanics, this belief is in principle true. However, in the real world exact knowledge of the *initial state* is not achievable. No matter how accurately the velocity of a particular particle is measured, one can demand that it be measured more accurately. Although we may, in general, recognise our inability to have such exact knowledge, we typically assume that if the initial conditions of two separate experiments are almost the same, then the final conditions will be almost the same. For most smoothly behaved systems, this assumption is correct. But for complex nonlinear natural systems, this assumption is far from the truth. At the turn of the 20$^{th}$ century, Henri Poincaré, another great French mathematician and natural philosopher, understood this phenomenon very precisely and wrote (as translated in *Science and Method* (1908, 1953)):

> A very small cause which escapes our notice determines a considerable effect that we cannot fail to see, and then we say that the effect is due to chance. If we knew exactly the laws of nature and the situation of the universe at the initial moment, we could predict exactly the situation of that same universe at a succeeding moment. But even if it were the case that the natural laws had no longer any secret for us, we could still only know the initial situation approximately. If that enabled us to predict the succeeding situation with the same approximation, that is all we require, and we should say that the phenomenon had been predicted, that it is governed by laws. But it is not always so; it may happen that small differences in the initial conditions produce very great ones in the final phenomena. A small error in the former will produce an enormous error in the later. Prediction becomes impossible, and we have the fortuitous phenomenon.

Indeed, this great French mathematician was working on a fortuitous phenomenon, as he called it, which was deep: *chaos*.

### 1.2 Rediscovering chaos: a new tool in the arsenal of science

In the movie *Jurassic Park,* Jeff Goldblum played a character who described himself as a "chaotician", an expert in chaos theory, dealing with "predictability in complex nonlinear systems…the "butterfly effect". As demonstration, he placed a drop of water on the back of Laura Dern's hand. "Which way is going to roll off?" he asked. She reasoned that a second drop, released at the same place as the first, would have the same path. To her surprise, each drop followed its own unique path rolling downward. "Why?" explained Goldblum, "…because tiny variations in the initial position and the skin never repeat and vastly affect the outcome… That is *chaos*."

James Gleick (1987) stated "where chaos begins classical science stops". As long as the world has had physicists inquiring into the laws of nature, it has suffered a special ignorance about the disorder in the atmosphere, in the turbulent sea, in the fluctuations in the ecological populations, in the erratic morphodynamic changes, in the beat of the heart and the pulsations of the brain. The irregular side of nature, the coexisting and switching dynamical regimes- these have been puzzles to science. But at the end of the nineteenth century J.Hadamard for the first time discovered chaos in a special (Hamiltonian) dynamical system called the geodesic flow on a manifold of negative curvature (Hadamard, 1898). Hadamard immediately understood the profound philosophical importance of his result: an arbitrarily small uncertainty on the initial condition entails a large uncertainty on the predicted state of the system after a sufficiently long time. Other scientists, such as P.Duhem and H.Poincaré also understood the importance of the phenomenon discovered by Hadamard, and Poincare (1908) discusses the relevance of sensitive dependence on initial condition to the dynamics of a hard sphere system, and to weather predictions. The early discovery of chaos had however no lasting influence on physics. The new ideas were forgotten and had to be rediscovered again, much later and independently. On the mathematical side, however, the work of Hadamard and Poincaré led to uninterrupted progress up to the present day, with contributions of such scientists as Kolmogorov, Smale, and many more. Incidentally, an essential step in the mathematical development of dynamical systems theory was the creation of *ergodic theory*, for which ideas originating in physics were important.

The time evolution of chaotic systems is typically complicated and irregular looking. Indeed, a regular (periodic or quasi-periodic) time evolution is predictable and therefore not chaotic. Although chaotic dynamical systems never exactly repeat, nor settle upon periodic trends, they are not random. They are deterministic in nature, they have structure, though subtle, and that makes them at least partially predictable. In other words, they show that the model is predictable in its unpredictability. When the interest for these kinds of irregular and complicated time evolutions of dynamical systems developed among physicists in the 1970s to give what is now called *chaos theory* (or theory of nonlinear dynamics more broadly), all kinds of new scientific tools existed that had not been available to Poincaré. One such tool is the electronic computer, which allowed Lorenz (1963) to compute in 1963 for the first time a chaotic time evolution of the simplified weather dynamics (see Chapter 3 for details), and to visualize it in the form of what we now call a *strange attractor*. Other tools were mathematical, like ergodic theory. Finally, there were new experimental laboratory tools permitting for instance a detailed study of the onset of hydrodynamic turbulence, one of the most challenging and

difficult phenomena. What we know about it results mostly from experimental studies, which knowledge is nowadays encapsulated into computational models. It is these experimental studies that showed that hydrodynamical turbulence is basically a deterministic chaos, as we would now say, corresponding to the claim of Ruelle and Takens (1971), that it is described by strange attractors. A mathematical proof of chaos in Navier-Stokes equations does not exist at this time (and when one is obtained, it will no longer create much excitement). We have thus here a very interesting situation from epistemological point of view, where we are firmly convinced of a certain mathematical fact (the existence of chaos in the solution of the inherently nonlinear Navier-Stokes equations) but our belief is based on experimental evidence, based on the data analysis of the observations.

As we mentioned above, the ultimate goal of physically-based modelling is actually *forecasting*, which raises another important issue related to deterministic chaos. Here the main objective is an attempt to provide a reliable forecast for some time into the future (the forecasting horizon) given some knowledge about the performances of the instantiated model and the situations (observations) in the real world system until the current time. The main issue is to set (adjust) the physically-based model to assimilate the initial (boundary) conditions at time *now* as accurately as possible, and to develop the forecasts up to the forecasting horizon making the best use of the observed available data. This brings up the notion of *data assimilation* techniques which can dramatically improve the performances of the mathematical model. State space data-driven models are the popular form of data assimilation into physically-based models (they are discussed in Chapter 5 of the thesis). Another alternative approach of improving the performance of physically-based models is to work with the differences (errors) between the model outputs and the observations. These differences provide useful insight of what is missing in the terms of processes and conceptualisation from the original model. The main idea is that the differences can be modelled very accurately using pure data-driven techniques such as various statistical methods, artificial neural networks, wavelet networks, fuzzy logic approximators and other techniques. The forecasts on the differences in a conjunctive use with the forecasts from the physically based model can lead to improvements in the results. *Data-driven modelling* is therefore a valuable complement to the physically-based modelling in the forecasting situations. However, data-driven modelling has also its own value independently of the physically-based modelling, and in the last decade has developed as an alternative to it. If one neglects the focus of using a mathematical model to better understand and describe the relationships between different variables and components of the underlying physical system, the main issue becomes the temporal accuracy of moment-to-moment estimates of the time series made by any model. In this respect, a data-driven model may give substantial forecasting improvements. Furthermore, this is particularly emphasised in the case where the physical processes are difficult to identify and formulate in an appropriate mathematical algorithmic form. Basically, what the data-driven model provides is a link (mapping) between the input-output sets of data observed on particular processes, such as meteorological forcing and water level in the sea. The data-driven model does not make certain assumptions and conceptualisations about the underlying processes that are connected in a “black-box” manner. The model has its own internal structure, for example, mimicking the brain structure as an artificial neural network does, and these

structures are very hard to interpret in connection to the physical processes being mapped in the model. Generally, the data-driven model has to be trained on data, namely observables of the natural processes. It can be said that the model has "learned" from data. This led to a new stream of modelling in hydroinformatics termed as model induction from data (Dibike, 2002). What is important here is that there should be completeness about the input data in relation to the physical processes so that an accurate and reliable model can be induced by the *learning machine* based on computational intelligence and machine learning techniques. In this way, the data-driven model naturally tries to minimise the dependency on knowledge of the real world processes.

Increasingly however, the *complementary* role of data-driven and mathematical modelling is being recognised. This is due to the fact that pure mathematical theories may fail to make accurate predictions of complicated water end environmental-related processes because the real world dynamical systems do not always obey equations with numerical and analytical solutions. Similarly, data-driven models induced from complicated observations and sometimes even missing observations of hidden processes are often inadequate because they fail to relate (and sometimes explain) complex effects from simple causes. It is only through the marriage of mathematical and data-driven modelling that many asserts of the complex dynamical processes can withstand reasonable tests. The theory of nonlinear dynamics and chaos is the good candidate to play this complementary role due to the fact that this theory originated based on mathematical analysis of deterministic dynamical systems described by a set of differential equations.

Thus, the main goal of this thesis is to assist in the development of this nonlinear literacy—or in the current context, numeracy- by describing, elaborating mathematically and illustrating the general principles and concepts that arise from modelling complex nonlinear dynamical systems in the aquatic environment. Our motivation and perspective is based on the considerable work in the last two decades that has been made in understanding nonlinear phenomena in the natural sciences (see, e.g. Campbell, 1989). In particular, the surge of interest in nonlinear dynamical systems and chaos theory has shown that such concepts as *bifurcations, attractors, basins of attraction, dynamical regimes, fractals, dimensions, predictability* and *local modelling* are essential for understanding the possible consequences of nonlinearity for modelling.

It must be said at this point that, however insightful and brilliant, the physical ideas of Poincaré on chaos were at the level of scientific philosophy. To some physicists chaos is a science of process rather than a state, of becoming rather than being. But now, after the rediscovery of chaos, science is focused and looking for examples of chaos which seem to be intrinsically inherited in many natural dynamical processes and systems. As Feigenbaum (1983), one of the greatest chaos theorists, put it: "Fifteen years ago, science was heading for a crisis of increasing specialisation. Dramatically, that specialisation has reversed because of chaos". Because it is a science of the global nature of nonlinear dynamical systems, it has brought together thinkers from fields that had been widely separated. Since new tools are now available for making strong claims about the universal behaviour of complexity, our present understanding of chaos in physics and mathematics is at the level of quantitative science. The theory of nonlinear dynamics and data analysis have progressed to the stage where most fundamental properties of

nonlinear dynamical systems have been observed in the laboratory and proven theoretically on various mathematical models.

What is currently lacking, and especially in the field of hydroinformatics, is the study of such nonlinear dynamical systems (e.g. movement of the body of water in oceans, rivers, subsurface and surface, ecological processes, hydrometeorological processes etc.) applying and extending the methods and techniques developed in the theory of nonlinear dynamics and chaos. Often we know little about the structure and interactions of such complex dynamical systems, but in practice we can measure (partly) its output and some of its inputs. In this respect, most direct link between the methods and concepts of deterministic chaos and the real world is the nonlinear analysis of data (time series) from real systems. Yet surprisingly the interactions on one level of understanding are often very similar to the interaction on other levels. Why is this so? Consider the following research questions: (i) Why do we find self-similar structures in biology and other disciplines? How does this relate to the self-similarity found in inanimate objects (phenomena) such as clouds, mountains, coast lines, turbulent eddies, fluid dynamics patterns and sedimentation patterns? Is there some way of generalising the notion of self-similarity to account for these types of phenomena? (ii) Is there a common reason why it is difficult to predict weather patterns, turbulence and other natural processes? Is this unpredictability due to limited knowledge of the underlying processes or is it somehow inherent in these complex systems? Can we quantify it? Can we increase our knowledge about the system under study in order to improve the mathematical models? (iii) How can we generally model and predict such unpredictable systems? The answers to these questions are apparently related to one simple fact: nature is chaotic.

Developing modelling methodologies and demonstrating applications of chaos to hydrology, hydrodynamics, meteorology, ecology (and other such disciplines), which are currently at very opening stage, may work towards reaching the level of quantitative science and creating tools for the engineering practice, which is the major objective of this thesis.

## 1.3 Scope and contributions

This work presents a novel hybrid modelling approach based on the theory of nonlinear dynamics and chaos. The modelling technique combines the multivariate phase-space reconstruction of the underlying dynamics based on time series of observables and mixture of local models learned in dynamic Bayesian network framework. The described modelling approach is applied for identification, modelling and prediction of hydrodynamical and hydrological systems: sea water level and surge dynamics along the Dutch cost, precipitation dynamics at De Bilt meteorological station in The Netherlands and rainfall-runoff dynamics of the Huai river in China. The results form these applications show that the methodology and the modelling framework presented in this thesis demonstrate reliable and accurate short-term forecasting performances and can be used as a modelling tool in the engineering practice.

Contributions of this thesis include the following:

A critical review of learning models from data from a statistical perspective focused on regression and density estimation, exemplifying both, the classical approaches based

on empirical risk minimisation and the approaches based on structural risk minimisation. Demonstration through examples and original discussions are provided.

Introduction, mathematical elaboration and demonstration of the methods and techniques based on the theory of nonlinar dynamics and chaos for the identification, reconstruction, delineation and quantification of the underlying dynamics of nonlinear dynamical systems from a time series of observables. The classical phase-space reconstruction of the dynamical systems known in the literature addresses methods and techniques based on univariate time series. This work further extends and proposes methodology for multivariate embedding, which is then tested and further demonstrated on the real case studies. Furthermore, this work elaborates how multivariate local models can be constructed in the reconstructed phase-space. Outlook of a methodology for analysis of spatially extended dynamical systems is provided.

Design, mathematical description, implementation and application of a novel data-driven modelling framework, termed as Hidden Markov Mixture of Models (experts). The framework aims at separating the seemingly complex global nonlinear dynamics into couple of local sub-dynamics that can be modelled by separate models (experts). The separate local multivariate models through a competition specialise on modelling different parts of the reconstructed phase space of the dynamical system where the gating procedure between the models is described with a dynamic Bayesian network expressed as hidden Markov model. First, this framework is tested using synthetic data generated by known dynamical systems and than applied to the case studies.

Development of methodology, based on the multivariate phase-space reconstruction of and the Shannon's conditional entropies for assessment of the local uncertainty and predictability of the dynamical system. Its application to the surge dynamics at Hoek van Holland tidal station in the North Sea.

The results from the applications of this novel hybrid modelling technique, which showed improved predictive performances in comparison with other nonlinear data-driven modelling techniques, such as artificial neural networks and fuzzy inference systems.

## 1.4 Thesis outline

This work is composed of seven chapters. A short overview of the material to be presented in the following chapters is given here.

Modelling nonlinear dynamical systems based on chaos theory is closely connected to data-driven modelling, as we will elaborate latter in this work. Chapter 2 describes the history and critically reviews learning from data, exemplifying both, the classical approaches based on empirical risk minimisation and the approaches based on structural risk minimisation. The problem of learning from data as an ill-posed problem is closely related to computational intelligence based on search and optimisation methods that are further discussed in this chapter.

Chapter 3 is at the heart of this work. It describes, elaborates mathematically and illustrates the main concepts of the theory of nonlinear dynamics and deterministic chaos. It further introduces and demonstrates the methods and techniques for the identification, reconstruction, delineation and quantification of the underlying dynamics of nonlinear

dynamical systems from a time series of observables. The phase-space reconstruction based on univariate time series is further extended and elaborated using multivariate embedding methodology proposed in this thesis. Finally, it elaborates how models can be constructed that realistically map the underlying structure dictating the dynamical evolution of the system.

Chapter 4 further extends this notion of models that learn from data by introducing the Bayesian network formalism. Special attention is given to dynamic Bayesian networks that are well suited for learning models from time series data observed on complex dynamical systems.

In Chapter 5 we propose, mathematically elaborate and demonstrate a novel hybrid framework for modelling nonlinear dynamical systems that draws on modelling based on both chaos theory and dynamic Bayesian networks.

Chapter 6 describes the results of the applications of the nonlinear dynamics and chaos to the following hydroinformatics problems: (i) Chaos and predictability of water levels and surges along the Dutch coast; (ii) Identification and reconstruction of the chaotic rainfall dynamics on different temporal scales and (ii) Rainfall-runoff modelling.

Chapter 7 summarises the conclusions drawn from this present work and highlights the strengths and weaknesses of the theory of nonlinear dynamics and deterministic chaos applied to hydroinformatics problems. It further identifies some related application areas that deserve further investigation in the future.

# Chapter 2
# Learning and Regularisation

## 2.1 General

Modelling nonlinear dynamical systems based on chaos theory and the methodology elaborated in this thesis is closely linked to data-driven modelling, i.e. learning models from data. This learning problem is an ill-posed problem and related to computational intelligence techniques based on search and optimisation. Thus, the main aim of this chapter is to provide a brief review of the learning theory from statistical and machine learning perspectives and the associated methods and techniques, which are relevant and further used in this work. It addresses both, the classical approaches based on Empirical Risk Minimisation (ERM) principle and the approaches based on Structural Risk Minimisation (SRM) principle. The problem of learning is so general that almost any question that has been discussed in statistical science has its analogy in learning theory. Furthermore, some important general results were first found in the framework of the learning theory and then reformulated and projected in the terms of statistics.

In the beginning of this thesis we postulated (without any discussion) that learning is a problem of *function estimation* on the basis of empirical data (observables). The ultimate goal is the *modelling* of a mapping $f: \boldsymbol{x} \rightarrow y$ from multidimensional input $x$ to output $y$. The output can be multidimensional, but we will mostly address situations and applications where it is a one dimensional real-valued vector. The multivariate function estimation is not, in principle, distinguishable from supervised machine learning. However, until recently supervised machine learning and multivariate function estimation, based on the statistical learning theory, had fairly distinct groups of practitioners, and small overlap in language, literature, and in the kinds of practical problems under study.

## 2.2 Setting of the learning problem

We describe the general model of learning from observables, based on Vapnik's statistical learning theory (Vapnik, 1995, 1998), through the following three components (Figure 2.1):

(i) A generator (G) of a random vectors $\boldsymbol{x} \in \boldsymbol{R}^n$, drawn independently from a fixed unknown probability distribution function $p(x)$.
(ii) A supervisor (S) who returns an output value $y$ to every input vector $\boldsymbol{x}$ (based on the true function $y=f(x)$), according to a conditional probability distribution function $p(y|x)$, also fixed and unknown.

(iii) A learning machine (LM) capable of implementing a set of functions ***f (x,*** $\alpha$***),*** $\alpha \in \Lambda$, where $\Lambda$ is a set of parameters[1].

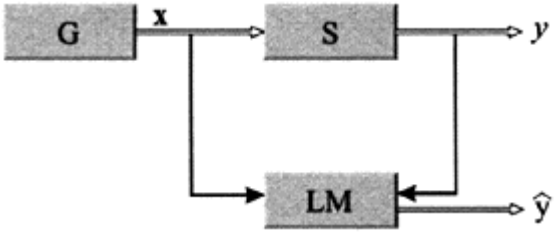



Figure 1.1. A model of learning from observables (Vapnik, 1995). During the learning process, the learning machine observes the pairs ***(x,*** *y)* (the training set) and uses them to adapt its parameters. The goal is to return a value *ŷ,* which is close to the supervisor's response y. After training, the machine should *generalise well,* that is, given a new input pattern *x,* the machine will provide a reasonable prediction of the unobserved output associated with this *x.*

In this manner, the learning problem is that of choosing from a given set of functions ***f(x,*** $\alpha$***),*** $\alpha \in \Lambda$, the one which best approximates the supervisor's response. The selection of the desired estimated function $(\hat{f}(\boldsymbol{x}) = \hat{y})$ is based on a training data set of *N* independent and identically distributed (i.i.d) observations drawn according to the joint probability distribution *p(x, y)=p(y|x)p(x):*

$D=(x_i, y_i)_{i=1\ldots N}$ (2.1)

[1] Note that the elements $\alpha \in \Lambda$ are not necessarily vectors, they can be any abstract parameters.

### 2.3 Learning and the problem of risk minimisation

In order to find the best available approximation to the system's (supervisor's) response, the *fit* of the model to the system is measured using a criterion representing the *loss* or discrepancy $L\left(y, \hat{f}(\boldsymbol{x}, \alpha)\right)$ between the response of the system $y$ to a given input $\boldsymbol{x}$ and the response provided by the learning machine. The performance of the model is measured by the expected value of the loss, termed as *expected risk:*

$$R(\alpha) = E_{x,y}\left(L(y, \hat{f}(\boldsymbol{x}, \alpha))\right) = \iint L(y, \hat{f}(\boldsymbol{x}, \alpha))\, p(x, y) dx dy \qquad (2.2)$$

The goal is to find the function $\hat{f}(\boldsymbol{x}, \alpha_0)$ minimises the risk functional $R(\alpha)$ (over the class of possible functions $f(\boldsymbol{x}, \alpha)$, $\alpha \in \Lambda$ in the situation where the joint distribution $p(x, y)$ is unknown and the only available information is contained in the training set (2.1). The quality $R(\alpha)$ represents the ability to yield good performance for all possible

situations (i.e. input patterns $(\boldsymbol{x}, y)$) and is thus called the *generalisation error*. The optimal set of parameters minimises the generalisation error:

$$\hat{\alpha} = \arg\min_{\alpha} R(\alpha) \qquad (2.3)$$

In order to minimise the risk functional, the following inductive principle can be applied (Vapnik, 1995):

(i) The risk functional $R(\alpha)$ is replaced by so-called *empirical risk* functional

$$R_{emp}(\alpha) = \frac{1}{N} \sum_{i=1}^{N} L(y, \hat{f}(\boldsymbol{x}, \alpha)) \qquad (2.4)$$

(ii) One approximates the function $\hat{f}(\boldsymbol{x}, \alpha_0)$ that minimises the risk (2.2) by the function $\hat{f}(\boldsymbol{x}, \alpha_N)$ minimising the empirical risk (2.4).

This corresponds to estimating the joint probability by the empirical density:

$$\hat{p}(x, y) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i)\, \delta(y - y_i),$$

where $\delta(\cdot)$ is the Dirac function. Minimising (2.4) is referred to as *training* the model. The data set $D$ and the empirical risk $R_{emp}(\alpha)$ are the *training set* and *training error*, respectively.

This principle is called the *empirical risk minimisation* inductive principle (ERM principle). An inductive principle defines a *learning process* if for any given set of

observations the learning machine chooses the approximation using this inductive principle. In learning theory the ERM principle plays a crucial role and is quite general. The classical methods for the solution of a specific learning problem, such as the least-squares method in the problem of regression estimation or the maximum likelihood (ML) method in the problem of density estimation, are realisations of the ERM principle for the specific loss (error) functions.

## 2.4 The three main learning problems

The formulation of the learning problems is rather broad and it naturally encompasses many specific problems. Generally applicable, the learning problems can be categorised in three main categories, namely: pattern recognition, regression estimation and density estimation. Further in this section we give a brief description of each of the learning problems on the basis of the model of learning from observations described in Section 2.2.

### *2.4.1 Pattern recognition*

Let the supervisor's output $y$ take only two values $y=\{0, 1\}$ and let the set of functions $f(\boldsymbol{x}, \alpha)$, $\alpha \in \Lambda$, be a set of indicator functions (binary type of functions which can take only two values: zero and one). For the pattern recognition problem the following loss function can be considered:

$$L(y, f(\boldsymbol{x},\alpha)) = \begin{cases} 0 \text{ if } y = f(\boldsymbol{x},\alpha) \\ 1 \text{ if } y \neq f(\boldsymbol{x},\alpha) \end{cases} \qquad (2.5)$$

For this loss function, the risk functional (2.2) determines the *classification error*. The problem, therefore, is to find an approximation function that minimises the probability of classification error when the join probability $p(x, y)$ is unknown, but the data (2.1) are given.

### *2.4.2 Regression estimation*

We now consider the case there the supervisor's answer $y$ is a real value, and when $\boldsymbol{f(x}, \alpha)$, $\alpha \in \Lambda$ is a set of real functions that contains the regression function:

$f(\boldsymbol{x}, \alpha_0)=\int y\, p(y|x)$.

(2.6)

It is known that the regression function is the one that minimises the risk functional (2.2) with the following loss function[2]:

$L(y, f(\mathbf{x},\alpha))=(y-f(\mathbf{x}, \alpha))^2$ (2.7)

Thus the problem of regression estimation is the problem of minimising the risk functional (2.2) with the loss function (2.7) in the situation where the joint probability $p(x, y)$ is unknown, but the data (2.1) are given.

### *2.4.3 Density estimation*

Finally, consider the problem of density estimation from a set of densities $p(\mathbf{x}, \alpha)$, $\alpha \in \Lambda$. For this learning problem we consider the following loss function:

$L(p(\mathbf{x}, \alpha))=-\log p(\mathbf{x}, \alpha)$. (2.8)

It is known that the desired density minimises the risk functional (2.2) with the loss function (2.8). Thus, again to estimate the density from the data one has to minimise the risk functional under the condition that the underlying probability distribution is unknown, but the i.i.d data (2.1) are given.

In the text above (Section 2.3) we mentioned that the empirical risk minimisation principle can be seen as a framework for the realisation of the classical methods for the solution of a specific learning problem, such as the least-squared method and the ML method. Indeed, by substituting the specific loss function for the regression estimation (2.7) in the empirical risk functional (2.4) one obtains the following functional to be minimised in order to find the proper model estimation and the optimal model parameters $\hat{f}(\mathbf{x},\alpha_0)$:

$$R_{emp}(\alpha) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - f(\mathbf{x},\alpha)\right)^2 \ , \qquad (2.9)$$

which forms the framework for the least-squared method. Alternatively, by substituting the loss function of the density estimation problem (2.8) in the empirical risk functional (2.4) one obtains the following functional to be minimised:

$$R_{emp}(\alpha) = -\frac{1}{N}\sum_{i=1}^{N}\ln p(\mathbf{x},\alpha) \ . \qquad (2.10)$$

[2] If the regression function $f(\mathbf{x})$ does not belong to $f(\mathbf{x}, \alpha)$, $\alpha \in \Lambda$, then the function $f(\mathbf{x}, \alpha_0)$ minimising the risk functional (2.2) with the loss function (2.7) is the closest to the regression in the metric $L_2(p): \rho\left(f(x), f(x,\alpha_0)\right) = \sqrt{\int\left(f(x)-f(x,\alpha_0)\right)^2 p(x)}$.

Minimising this functional is equivalent to the maximum likelihood method.

## 2.5 The paradigm of solving learning problems based on the Empirical Risk Minimisation principle

The setting of the learning problem involves two major requirements: (i) to estimate the desired function from a wide set of functions and (ii) to estimate the desired function on the basis of a limited number of examples (observables). The methods developed in the framework of the classical learning paradigm (created in the 1920s and 1930s) did not take into account these requirements. Therefore, in the 1960s considerable effort was put into both the generalisation of the classical results for wider sets of functions and the improvement of the existing techniques for statistical inference. Although there are several classical techniques for estimating the parameters of a set of functions and density estimations, such as the method of moments (dated back to Johan Bernoulli, 1667–1748), method of maximum likelihood (Fisher, 1920), method of least-squares (dated back to Gauss 1777–1855), method of minimum cross entropy (Shanon, 1949), method of Bayesian estimation (dated back to Bayes, 1763), method of probability weighted moments (Greenwood et al., 1979) and method of L-moments (Hosking, 1990), most of the models of function estimation are based on the maximum likelihood method. It forms an inductive engine in the classical paradigm. Textbooks such as Benjamin and Cornell (1970) and Berger (1985) treat the classical methods in details.

### *2.5.1 Maximum likelihood (ML) and the density estimation problem*

It is difficult to trace back who introduced the ML method, though Daniel Bernoulli (1700–1780) was one of the first to report it. In 1922 Fisher developed the ML method for estimating the unknown parameters of the density (Fisher, 1952). The method can be summarised as follows: Let $p(\boldsymbol{x}, \alpha)$, $\alpha \in \Lambda$, be a set of density function where in this case the set $\Lambda$ is necessarily constrained in $\boldsymbol{R}^n$ ($\alpha$ is a $n$-dimensional vector). The unknown density $p(\boldsymbol{x}, \alpha_0)$ belongs to this set. The problem is to estimate this density using i.i.d. data $(x_1,\ldots, x_N)$ distributed accordingly to this unknown density. Fisher suggested approximating the unknown parameters by the values that maximise the function:

$$L(\alpha) = \sum_{i=1}^{N} \ln p(x_i, \alpha). \qquad (2.11)$$

The ML gives an asymptotically unbiased parameter estimation, and of all the unbiased estimators it has the smallest mean squared error. The variances approach asymptotically to:

$$Var(\alpha) = -E(\partial^2 \ln L(\alpha)/\partial\alpha^2) \qquad (2.12)$$

Furthermore these estimators are invariant, consistent and sufficient (Hald, 1952). Analytical expressions for the parameter estimation are sometimes difficult to derive, which means that numerical optimisation routines have to be derived in order to determine the maximum of the likelihood function. However, those numerical routines may also have problems in finding the optimum due to the reason that the likelihood function can be extremely flat for large sample sizes and due to the existence of local maxima.

Some of the characteristics of the ML estimators discovered during their applications in the past decades are: (i) ML methods are straightforward to implement; (ii) ML estimators may not exist (Vapnik, 1995), and when they do, they may not be unique or give a biased error (Koch, 1991); (iii) ML estimators may give inadmissible results (Lundgren, 1987); (iv) the likelihood function can be used for other purposes than just finding the parameters: values close to the ML are more plausible than those further away. This argument can be utilised to obtain an interval, which comprises a plausible range of values for certain parameters α; (v) ML estimators are adaptable for more complicated modelling situations, because ML satisfies a convenient invariance property (Huber, 1964): If $q=f(\alpha)$, where $f$ is an objective function, then $q_{ML}=f(\alpha_{ML})$. Thus having found ML estimators for one parameterisation, the ML estimators for other parameterisations are immediate.

Furthermore, the ML method allows the linking of the risk function (2.2) and the assumption on the noise distribution on the observed output; see Section 2.5.3 on regression estimation model. One can say that the ML is very useful, since it is quite straightforward to evaluate from the ML estimators and the observed information. Nonetheless, it is an approximation and should only be trusted for large data sets (though the quality of approximation will vary from model to model).

### *2.5.2 ML and the pattern recognition (discriminant analysis) problem*

Using the ML technique, Fisher (1922) considered a problem of pattern recognition (which he called discriminant analysis). He proposed the following model:

> There exist two categories of data distributed according to the two different statistical laws $p_1(x, \alpha^*)$ and $p_2(x, \beta^*)$ (densities, belonging to parametric classes). Let the probability of occurrence of the first category of data be $q_1$ and the probability of the second category be $1-q_1$. The problem is to find a decision rule that minimises the probability error.

Knowing these two statistical laws and the value of $q_1$, one can immediately construct such a rule: The smallest probability of error is achieved by the decision rule that considers vector $x$ as belonging to the first category if the probability that this vector belongs to the first category is not less then the probability that this vector belongs to the second category. This happens if the following inequality holds:

$$q_1 p_1(x, \alpha^*) \geq (1-q_1) p_2(x, \beta^*). \tag{2.13}$$

One can consider this rule in the equivalent form as:

$$f(x) = sign\left\{\ln p_1(x,\alpha^*) - \ln p_2(x,\beta^*) + \ln\frac{q_1}{(1-q_1)}\right\}, \qquad (2.14)$$

called the discriminant function (rule), which assigns the value of 1 for representatives of the first category and value of −1 for representatives of the second category. To find the discriminant rule one has to estimate two densities $p_1(x, \alpha^*)$ and $p_2(x, \beta^*)$. In the classical paradigm ML method in the framework of the ERM is used to estimate the parameters $\alpha^*$ and $\beta^*$ of these densities.

### *2.5.3 ML and the regression estimation model*

Regression estimation in the classical paradigm is based on another model, the so-called model of measuring a function with additive noise:

Suppose that the unknown function has a parametric form:

$$f_0(x)=f(x, \alpha_0) \qquad (2.15)$$

where $\boldsymbol{x}$ can be a multivariate vector and $\alpha \in \Lambda$ is an unknown vector of parameters. Suppose also that in any point $x_i$ (pattern in a multidimensional space) one can measure the value of this finction with additive noise:

$$y_i=f(x_i, \alpha_0)+\varepsilon_i, \qquad (2.16)$$

where the noise εi does not depend on xi and is distributed according to a known density function $p(\varepsilon)$. Then the problem is to estimate the function $f(\boldsymbol{x}, \alpha_0)$ from the set $f(\boldsymbol{x}, \alpha)$, $\alpha \in \Lambda$, using the data obtained by measurements of the function $f(\boldsymbol{x}, \alpha_0)$, corrupted with additive noise.

In this model, using the observations of pairs $D=(x_1, y_1),\ldots, (x_N, y_N)$ one can estimate the parameters $\alpha_0$ of the unknown function $f(\boldsymbol{x}, \alpha_0)$ by the ML method, namely by minimising the functional:

$$L(\alpha) = \sum_{i=1}^{N} \ln p\left(y_i - f(x_i,\alpha_0)\right), \qquad (2.17)$$

where $p(\varepsilon)$ is a known function and $\varepsilon=y-f(\boldsymbol{x}, \alpha_0)$. Under the assumption of normal distribution law:

$$p(\varepsilon) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right) \qquad (2.18)$$

with zero mean and some fixed variance $\sigma^2$ (scalar in univariate or covariance matrix in multivariate case) as a model of noise, one can obtain the likelihood of the data set, which is the least-squared method:

$$L^*(\alpha) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y_i - f(x_i,\alpha)\right)^2 - N\ln\left(\sqrt{2\pi\sigma}\right). \qquad (2.19)$$

Maximizing the likelihood (2.19) over the parameters α is equivalent to minimising the function:

$$M(\alpha) = \sum_{i=1}^{N}\left(y_i - f(x_i,\alpha)\right)^2 \qquad (2.20)$$

which is the so-called least-squared functional, where the loss function (2.7) is based on the Euclidean norm $\|\cdot - \cdot\|_2$. The least-squared solution in this case is a special case of the empirical risk minimisation inductive principle. Choosing other laws $p(\varepsilon)$, one can obtain other ML parameter estimators in the regression problem (see Huber, 1964 for details).

### *2.5.4 Noisy output and the generalisation error*

In the regression estimation problem, we have shown the link between the assumed output noise distribution and the loss (error) function. Let us now briefly demonstrate the influence of this noise on the generalisation performance using the expected risk functional. Let us assume again that the system is corrupted by additive, independent noise, with zero mean and $\sigma^2$ variance: $y=f(\mathbf{x}, \alpha)+\varepsilon$. The underlying joint probability distribution, which generates the output of the system, can be written as $p(x, y)=p(y|x)\,p(x)=p(\varepsilon)p(x)$. Substituting the loss function (2.7) into (2.2), for the expected risk (generalisation error) follows:

$$R(\alpha) = \int_{\mathbf{x}}\int_{\varepsilon}\left(f(\mathbf{x},\alpha)+\varepsilon - f(\mathbf{x},\alpha_0)\right)^2 p(x)\,p(\varepsilon)\,d\varepsilon\,dx \qquad (2.21)$$

By recaling that $\int p(\varepsilon)d\varepsilon = 1$, $\int \varepsilon p(\varepsilon)d\varepsilon = 0$ (zero mean), and $\int \varepsilon^2 p(\varepsilon)d\varepsilon = \sigma^2$, one can derive the following expression for the generalisation error:

$$R(\alpha) = \sigma^2 + \int_{\mathbf{x}}\left(f(\mathbf{x},\alpha) - f(\mathbf{x},\alpha_0)\right)^2 p(x)\,dx\ . \qquad (2.22)$$

The difference in the generalisation error between the noisy and noise-free case is an additive constant. This gives the following insights:

- The noise level is a lower bound on the generalisation error
- The generalisation error of a perfect model learned from the data is the variance of the output noise (the integral on the right-hand side of Equation 2.22 vanishes)

• Output noise can be neglected as far as the generalisation error is concerned. However, this of course is not the case with the empirical risk (training error).

### *2.5.6 Linear regression*

In this part we describe the particular case where the model is linear, since it will be used latter for constructing the local linear models in the phase space of a dynamical system (see Section 3.3.7). The unknown function in a parametric form is given as:

$f(\boldsymbol{x}, \alpha_0)=\mathbf{x}^{\mathrm{T}}\cdot\alpha$ (2.23)

where $^{\mathrm{T}}$ is transpose operator and $\alpha \in \Lambda$ is an unknown vector of parameters. We also assume that the system is linear, corrupted by additive independent normal noise. The data set consists of a number of $N$ input-output pairs, which are mapped as: $y=x\cdot\alpha_0+\varepsilon$. The goal is to estimate the optimal set of parameters $\alpha_0$. Let us denote the by **X**, **Y** and **E** the $N\times P$, $N\times 1$ and $N\times 1$ matrices (respectively) containing the transposed input, output and noise vectors:

$$\mathbf{X}=\begin{bmatrix} x_1^{\mathrm{T}} \\ x_2^{\mathrm{T}} \\ \vdots \\ x_N^{\mathrm{T}} \end{bmatrix} \quad \mathbf{Y}=\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad \mathbf{E}=\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{bmatrix} \tag{2.24}$$

The empirical risk functional is expressed simply as:

$$R_{emp}(\alpha)=\frac{1}{N}\left\|\mathbf{X}\cdot\alpha-\mathbf{Y}\right\|^2 \tag{2.25}$$

The linear maximum likelihood estimator is obtained by minimising the empirical risk (2.25). Taking the derivative $\nabla R_{emp}(\alpha)=\frac{2}{N}\mathbf{X}^{\mathrm{T}}\left(\mathbf{X}\cdot\alpha-\mathbf{Y}\right),$ one can obtain the well-known expression of the linear regression estimator:

$$\hat{\alpha}=\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y} \tag{2.26}$$

### *2.5.7 Nonlinear regression*

For the more general case of a nonlinear regression parameterised model $f(\boldsymbol{x}, \alpha_0)$, using the squared loss (error) function, the empirical risk is expressed as in (2.9). Unlike the linear case, there is no analytical solution for the minimisation of the empirical risk. Finding the proper model estimation and the optimal model parameters $\hat{f}(\boldsymbol{x}, \alpha_0)$ is a

standard optimisation problem (see Section 2.8 for details). The gradient of the empirical risk is:

$$\nabla R_{emp}(\alpha) = \frac{2}{N}\sum_{i=1}^{N} \mathbf{J}_f(\alpha)\left(y_i - f(\boldsymbol{x}_i, \alpha)\right) . \qquad (2.27)$$

For multivariate model *f,* $\mathbf{J}_f$ is the Jacobian matrix calculated in α. For an univariate model, $\mathbf{J}_f = \nabla f(\boldsymbol{x}, \alpha)$.

## 2.6 Non parametric methods of density estimation

Estimating densities from some narrow set of densities or so-called parametric set of densities (e.g. from a set of densities determined by a finite number of parameters) was the subject of the classical paradigm, where a "self-evident" type of model inductive engine (e.g. ML method) was used. To estimate a density from the wide (nonparametric) set one required a new type of inference that contains regularisation techniques. Regularisation, loosely speaking, means that while desired model is constructed to map approximately the observed vectors to the observed output of the system, constrains are applied to the construction of the model with the main goal of reducing the expected risk (generalisation error). We will return to the important subject of regularisation further in this chapter. At the beginning of 1960s several such types of (nonparametric) algorithms were suggested (Rosenblatt, 1956; Parzen, 1962; Chentsov, 1963). In the middle of 19970s the general approach for creating for creating these kinds of algorithms was found (Vapnik and Stefanyuk, 1978). Nonparametric methods of density estimation gave rise to statistical and machine learning algorithms that overcome the limitations of the classical methods.

### *2.6.1 Parzen's windows method*

Among the various nonparametric methods of density estimation, the Parzen windows method (Parzen, 1962) probably is the most popular and attractive. According to this method, one first has to determine the so-called *kernel function*. For simplicity we consider here a simple kernel function:

$$K(x, x_i; \gamma) = \frac{1}{\gamma^n} K\left(\frac{x - x_i}{\gamma}\right), \; x \in R^n , \qquad (2.28)$$

where *K(u)* is a symmetric unimodal density function. Using this function and the ERM principle, one can determine the density estimator as:

$$p(x) = \frac{1}{N}\sum_{i=1}^{N} K(x, x_i; \gamma) . \qquad (2.29)$$

In the 1970s a comprehensive asymptotic theory for Parzen-type of nonparametric density estimation was developed (Devroye, 1985). This theory includes the following two important assertions:

(i) Parzen's estimator is consistent (in various metrics) for estimating a density from a wide class of densities (functions)
(ii) The asymptotic rate of convergence for Parzen's estimator is optimal for "smooth" densities.

The main drawback of the findings was that for both classical models (pattern recognition and regression estimation) using nonparametric methods instead of parametric methods, it is possible obtain a good approximation to the desired dependency if the number of observation is *sufficiently large*. Naturally a question follows: What does a sufficiently large data set means? This question will be further addressed in this chapter with the description of the *structural risk minimisation* principle.

### *2.6.2 The problem of density estimation is an ill-posed problem*

Let us recall that the learning from data or simply the learning problem is to obtain a function f in a given set $\Lambda$ that minimises the risk functional (generalisation error):

$$R(\alpha)=\int L(x, \alpha)p(x)dx \tag{2.30}$$

Let us now focus on the problem of estimating the density $p(x)$. If one can estimate this density correctly, one could hope in turn to estimate the $R(\alpha)$. We now wish to solve the probability distribution problem, i.e. find the density $p(u)$ (if it exists) satisfying the integral equation:

$$\int_{-\infty}^{x} p(u)du = P(x), \quad \forall x \tag{2.31}$$

where $P(x)$ is an unknown probability distribution function, but we have number $N$ of observations $x_1,\ldots x_N\ldots$ available, sampled from this distribution. The unknown p.d.f. can then be approximated by using the empirical distribution function (Figure 2.2):

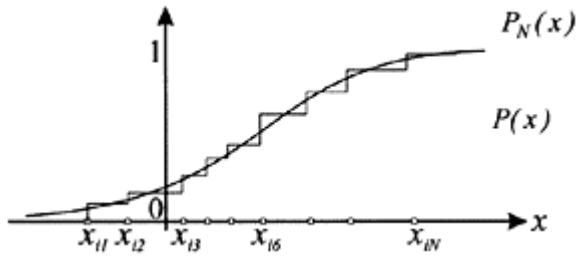



Figure 2.2. The empirical distribution function $P_N(x)$ constructed from the

observed data $x_1,\ldots x_N$ approximates the truth probability distribution function $P(x)$ (Vapnik, 1998).

$$P_N(x) = \frac{1}{N}\sum_{i=1}^{N} H(x - x_i) \ , \tag{2.32}$$

where $H$ is the Heavyside (step) function. Its derivative is the Dirac function $\delta$. According to the fundamental Glivenko-Cantelli theorem (Glivenko, 1933), the empirical distribution function (2.32) converges uniformly towards the desired function $P(x)$. The approximation problem of density estimation then becomes:

$$\int_{-\infty}^{x} p_N(u)du = P_N(x), \quad \forall x \tag{2.33}$$

where the obvious solution to this problem is expressed as:

$$P_N(x) = \frac{1}{N}\sum_{i=1}^{N} \delta(x - x_i) \tag{2.34}$$

the empirical density estimation. Therefore, one has to solve the integral equation (2.31) for the case where instead of the exact right-hand side, one knows an approximation that converges uniformly to the unknown function as the number of observation increases. Despite the (uniform) convergence of $P_N(x)$ towards $P(x)$, the solution $p_N(u)$ of (2.33) does not converge towards the (unknown) solution $p(u)$ of (2.31). The density estimation problem is thus *ill-posed* problem (there may be a continuum of solutions in a wide class of functions $\{p(u)\}$ for a particular data set). Notice that the use of the empirical density (2.34) as an estimate of $p(x)$ in (2.3) leads to the expression of the empirical risk or (unregularised) training error.

In order to practically illustrate the fact that the density estimation is an ill-posed problem, we consider a classical example in nonlinear regression. Consider the extremely simple setting: we try to estimate a sinusoid on 10 points with $x$ values generated in the interval [0;2] and y values in [−1;1].

The model is a simple one-parametric function $y=sin(\alpha\pi x)$. It is a nonlinear model and depends on a single parameter $\alpha$, which in this case represents the frequency. Let us consider for example that we generate 10 observations from this sinusoidal with parameter $\alpha=1$, which are slightly polluted by an additive independent with noise (see Figure 2.3). The noise level is rather low with a zero mean and variance 0.03, which gives us a signal-to-noise ratio of 5.1%.

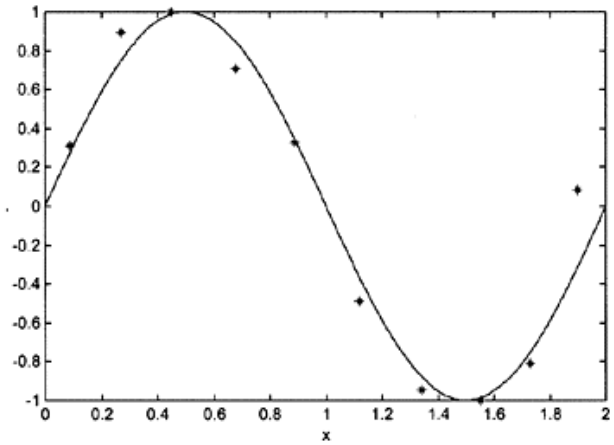



Figure 2.3. The 10 sampled data points polluted with additive independent white noise (variance 0.03) are displayed with the sinusoid from which they were sampled.

In this example the underlying mapping is taken as one of the possible models (from a wide set of functions). The best estimation of the parameter (i.e. the one that gives minimum risk and generalise best) would obviously be $\hat{\alpha} = 1$. In this particular problem where the model depends on a unique parameter, it is unnecessary to route to multidimensional optimisation (to be discussed in this chapter). A simple line search will suffice. The loss function, expressed as a mean squared error, is a function of a single parameter:

$$L(\alpha) = MSE = \frac{1}{10}\sum_{i=1}^{10}\left(y_i - \sin(\alpha\pi x)\right)^2 \qquad (2.35)$$

Figure 2.4 shows the behaviour of the MSE as a function of the parameter α, which was varied on the interval [0;15] using the step of 0.01. As expected the MSE (value 0.0158) reaches its first minimum around $\hat{\alpha} = 1$ (1.04 precisely), which is only a good estimated local minimum, since the global minimum is in $\hat{\alpha} = 10$, where the MSE actually reaches value 0, due to the Ocam's razor (to be discussed latter). The resulting model for the best MSE and the optimal parameter $(\hat{\alpha} = 10)$ is plotted together with the underlying mapping function and the observables on Figure 2.5. The empirical risk minimisation principle indeed minimises the distance to the data, as these points are actually on the model.

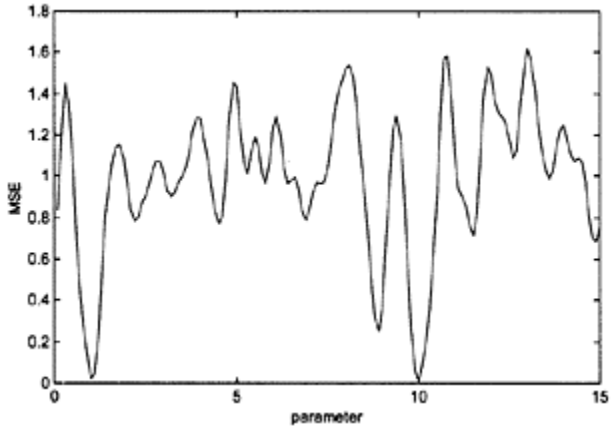



Figure 2.4. The MSE as a function of the model parameter α (two minima are clearly visible).

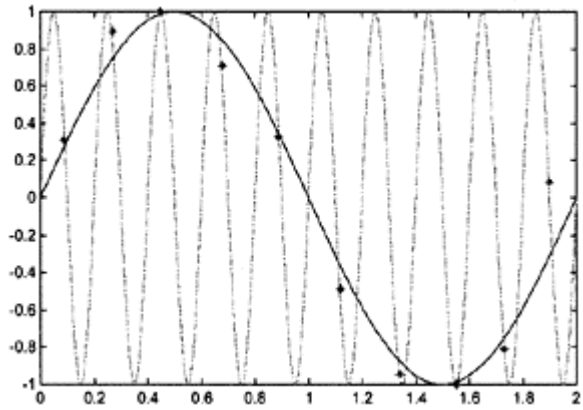



Figure 2.5. The solution function which resulted from the empirical risk minimisation principle (dashed) together with the observed data points and the “true” underlying mapping (solid line).

It is clear that this solution is not a favourable one in terms of generalisation capabilities. Apart from a couple of observables where both curves cross each other, the model produces estimations irrelevant to the underlying mapping. The reason for this is simple: whatever set of N points we observe (generate in this case) with ordinates in [−1;1], there exists a value of model parameter α, such that the associated sinusoid approximates the data arbitrarily closely. This is reflected in the fact that the sinusoid, even thought it has a single parameter, has *infinite capacity*, that is, it can interpolate with arbitrary precision any set of any number of points within its range.

In order to create a better model, some background or *a priori* information (knowledge) has to be presented to this simple learning machine. Let us add a small *regularisation term* to the loss function (the mean squared error): $C(\alpha)=0.01\alpha^2$. This regularisation term corresponds to imposing a penalty on the large values of the parameter α. In other words, we express our belief that the probability density of the parameter α, should be more densely distributed towards small values, which favours low frequencies, i.e. smooth functions. The resulting loss function expressed in a term of MSE as a function of parameter α is shown on Figure 2.6. It is now clearly visible that due to the regularisation effect, there exists only one clear global minimum of the MSE at the value of the parameter $\hat{\alpha} = 1.04$. Figure 2.7 displays the shape of the resulting model $\hat{y}=sin(1.04\pi x)$, original mapping and the data points. Note that despite the limited amount of data available, the nonlinear model provides a fairly good approximation of the underlying mapping in the domain of the data.

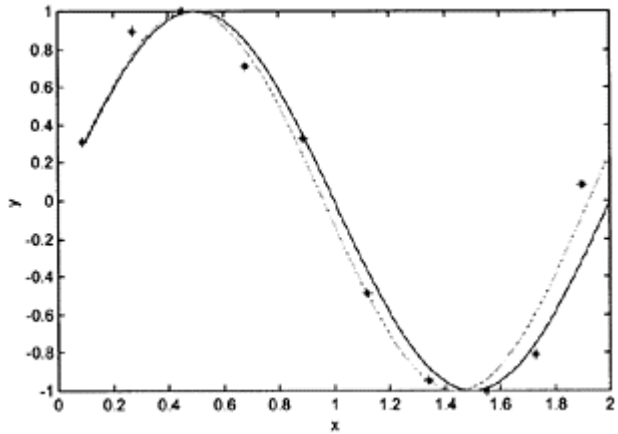



Figure 2.6. The MSE as a function of the model parameter α with the help of the regularisation term, resulting in one clear MSE minimum (at $\hat{\alpha} = 1.04$)

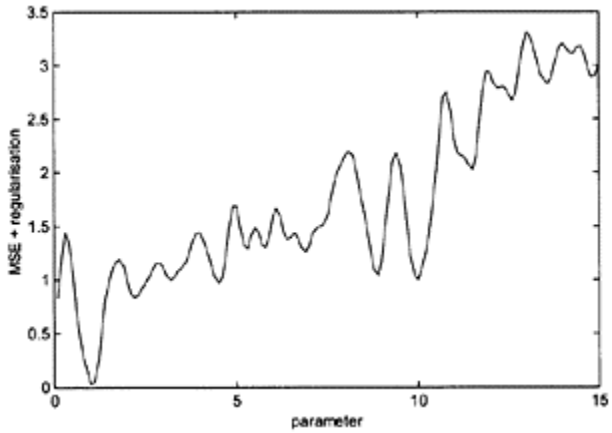



Figure 2.7. The solution function which resulted from the regularised empirical risk minimisation principle (dashdotted) together with the observed data points and the "true" underlying mapping (solid line).

### *2.6.3 Artificial neural networks*

The Artificial Neural Network (ANN) approach to nonlinear regression and density estimation is a computational learning approach inspired by studies of the brain and nervous systems in biological organisms. The inspiring functionality of a biological neural system has been attributed to the parallel-distributed processing nature of the biological neurons. An ANN emulates this structure by distributing computations (learning tasks) to small and simple processing units, called artificial neurons, which are interconnected to form a connectionist model—network (see Figure 2.8). The historical developments of the first ANN-type of learning machine point back to Rosenblatt (1962) who suggested the first model of perceptron. He described the model as a program for computers and demonstrated with simple pattern recognition experiments that this model can be generalised. In 1986 several authors independently proposed a method for simultaneously constructing the vector coefficients for all neurons of the perceptron model using the so-called *back-propagation method* (LeCun, 1986; Rumelhart, Hinton and Williams, 1986), which was one of the important milestones in the general learning theory.

Let us now return to our learning problem and briefly describe the ANNs as nonlinear nonparametric regression estimators. The regression function ($f: \boldsymbol{x} \rightarrow y$) is a multivariate nonlinear and especially time-varying (dynamical) mapping, which is of particular focus of this thesis. When the exact nonlinear underlying structure of this mapping cannot be establish *a priori*, the general estimator may be synthesised as a combination of parametrised basis functions:

$$\hat{f}_t = (\boldsymbol{x}_t, \boldsymbol{\theta}_t) = G_{t,k}\left(\boldsymbol{\theta}_{t,k}; \left( \ldots \sum_j G_{t,j}\left(\boldsymbol{\theta}_{t,k}; \sum_i G_{t,i}(\boldsymbol{\theta}_{t,k}; \boldsymbol{x}_t)\right)\ldots\right)\right) \tag{2.35}$$

where $G_{t,i}(\boldsymbol{x}_t, \boldsymbol{\theta}_{t,i})$ denotes multivariate basis function and $\boldsymbol{\theta} \in R^m$ is a set of model parameters. These multivariate basis functions may be generated from univariate basis function using radial basis, tensor product, wavelet basis or ridge construction methods. This type of regression is often referred to as "nonparametric" due to the large number of the basis functions. Equation (2.35) encompasses a large number of nonlinear estimation methods such as: projection pursuit regression (Fridman and Stuetzle, 1981; Huber 1985), Volterra series (Billings, 1980; Mathews, 1991), fuzzy inference systems (Jung and Sun, 1993), generalised linear models (Nelder and Wedderburn, 1972), multivariate adaptive regression splines (MARS) (Friedman, 1991) and many *artificial neural networks* paradigms including functional link networks (Pao, 1989), multi-layer perceptrons (MLPs) (Rumelhart et al., 1986), radial basis function networks (RBFs) (Moody and Darken, 1988; Lowe, 1989; Poggio and Girosi, 1990), wavelet networks (Zhang, 1993; Bakshi and Stephanopoulos, 1993; Juditsky 1997) and hinging hyper-planes (Breiman, 1993). For an introduction to ANNs we refer to any of the following textbooks: (Bishop, 1995; Haykin, 1999; Hecht-Nielsen, 1990; Ripley, 1996).

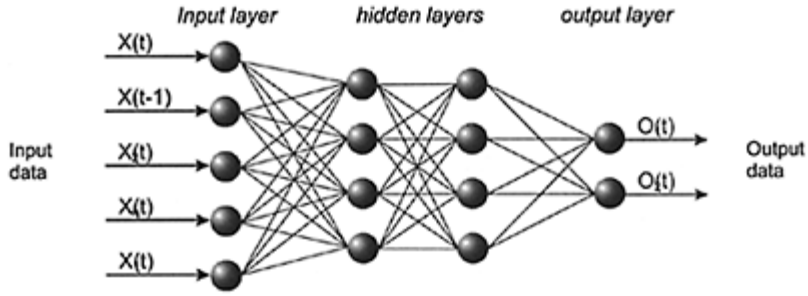



Figure 2.8. Typical multi-layer perceptron architecture.

ANNs in data-driven modelling are interesting for several reasons: (i) they provide a convenient generic non-linear modelling tool to the practitioners; (ii) they can approximate any continuous function arbitrarily well as the number of neurons (basis functions) increases without bound (Cybenko, 1989; Hornik, 1989; Poggio and Girosi, 1990); (iii) they have been successfully applied to many complex practical problems, including speech recognition (Robinson, 1994), hand-written digit recognition (Le Cun et al., 1989), financial modelling (Refenes, 1995), medical diagnosis (Baxt, 1990) among others, and finally to many civil engineering problems ranging from rainfall-runoff modelling in hydrology (Minns, 1995), runoff modelling (Minns 1998), ocean water level forecasting (Frison et.al, 1994, Abarbanel, 1996), storm surges classification (Zijderveld, 2003), sediment transport modelling (Bogaard 2000), automated land-cover image classification (Velickov et al., 2000) to geological classification and regression (Alvarez,

2001). In this work we will make use of three types of neural network architectures in some of the practical applications (see Chapter 6): fixed dimension MLPs, wavelet networks, and both fixed and variable dimension RBFs.

MLPs among the other ANNs architectures have enjoyed a privileged position in the research community because of their simplistic structure, easy algorithmic implementation, approximating capabilities, relation to the biological systems and various historical reasons. Figure 2.8 shows a typical two hidden layer MPL with logistic sigmoid basis functions in the hidden layers and a single output linear neuron. Mathematically, networks of this type can be expressed as:

$$\hat{f}_t = (x_t, \theta_t) = \sum_i \theta_{t,i} \left( \sum_k \sigma \left( \theta_{t,k} \left( \dots \sum_j \sigma \left( \theta_{t,j} \sum_i \sigma(\theta_{t,i} x_t + b_{t,i}) + b_{t,j} \right) \dots \right) + b_{t,k} \right) \right) + b_{t,l} \qquad (2.36)$$

where $\boldsymbol{b}_{t,i}$ denotes the bias on the ith neuron in the first layer and $\boldsymbol{\theta}_{t,i}$ is a row vector containing the weights connecting each input pattern with the ith neuron. The transfer (basis) function $\sigma$ in the input and hidden layers is usually a nonlinear, increasing, bounded function such as the hyperbolic tangent (tanh), the error function (erf) or the simple sigmoid function: $\sigma(u)=1/(1+\exp(-u))$. For regression estimations, the transfer function at the output layer is usually kept linear (for extrapolation reasons), while for pattern recognition problems it is customary to apply a nonlinear bounded function again (for soft classification purposes). This allows the interpretation of the output of the network as a class membership. The choice of the number of input and output units is generally problem and process dependent. In the time series modelling and nonlinear dynamic system identification applications in this thesis, we will mostly have one output (the forecast) and as many inputs as necessary for proper reconstruction of the dynamics of the systems analysed. Although the MLPs discussed in this thesis exhibit a feed-forward architecture, recurrent and modular type of architectures (Vassilios and Kehagias, 1998) have also been applied to some of the analysed problems (see Chapter 6). A detailed description of the various ANN architectures is beyond the scope of this thesis.

Finally, we would like to stress that almost fifteen years have passed since the construction of the first efficient ANN-type of learning machine. From a conceptual point of view, important achievements were made in constructing and investigating different structures of ANNs. In spite of consequent achievements in some applications using ANNs, the theoretical results obtained did not contribute much to the general learning theory. The so-called overfitting phenomenon observed in experiments is actually a phenomenon of “false structure” known in the theory for solving ill-posed problems (Denker et al., 1987). From the theory for solving ill-posed problems, regularisation techniques were adopted that prevent overfitting (Plaut at al., 1986; Krogh and Hertz, 1992), force structural optimisation of ANNs (e.g. optimal brain damage and optimal brain surgeon) (Le Cun et al., 1990; Hassibi and Stork, 1993) and stop training early (Ljung et al., 1992).

## 2.7 Regularisation

In the previous section, we described and demonstrated that the induction of models from data using the ERM principle is ill-posed problem. In the 1960s and 1970s, in various branches of mathematics, several streams of investigation were developed which founded the basis for *regularisation theory* for solving ill-posed problems. We introduce the concepts of well-posed and ill-posed problems and the regularisation technique in a general context, which become very important for creating new paradigms in solving learning problems.

### *2.7.1 Well-posed and ill-posed learning problems*

The existence of the ill-posed problems has been observed in the early 1890s by the French mathematician Hadamard (1902) who considered a typical *inverse problem*, the problem of solving operator equations (finding *f* that satisfies the equality):

$$Af = F, \quad F \in \mathcal{F} \tag{2.37}$$

where $A$ is an operator and F belongs to the metric space $\mathcal{F}$. $A$ can be a linear as well as a nonlinear operator. Typical examples include derivative or integral operators. For example, a system governed by a second order differential equation can be discretised and expressed as a linear equation $\boldsymbol{A}\cdot f=F$, where $F$ is a set of discrete measurements and $\boldsymbol{A}$ is a known matrix representing the differential equation f (e.g. the second derivative is expressed simply as a band diagonal matrix with −2 on the diagonal and 1 on the upper and lower first band). In the context of the parametric regression estimation, $F$ is the observed data, and $f$ is the unknown data model, containing a set of parameters. This is a typical inverse problem as we wish to invert the *cause+system=effect* type of dependency. Knowing the cause and the effect, we try to reason about the system, or more precisely about the processes underlying the system.

Hadamard noticed that in some cases, equation (2.37) is ill-posed: a small deviation on the right-hand side of this equation ($F_\delta$ instead of $F$, were $\|F-F_\delta\|<\delta$ is arbitrarily small) can result in a large deviation in the solutions $f$. In the case where the right-hand side of the equation is not exact, the functions $f_\delta$ that minimise the risk functional

$$R(f)=\|\boldsymbol{A}f-F\delta\|^2 \tag{2.38}$$

does not guarantee a good approximation to the desired solution, even if $\delta\rightarrow 0$. In the middle of the 1960s (Tikhonov, 1963) it was discovered that if instead of minimising the function (2.38) one minimises the functional

$$R^*(f)=\|\boldsymbol{A}f-F\delta\|^2+\gamma(\delta)\Omega(f) \tag{2.39}$$

where $\Omega(f)$ is so-called regularisation functional, and $\gamma(\delta)$ is an appropriately chosen parameter (depending on the level of noise), then it is possible to obtain a solution that converges to the desired one as $\delta$ tends to zero (Tikhonov, 1963; Ivanov, 1962; Phillips

1962). The regularisation parameter γ, also called *hyper-parameter*, implicitly defines a structure on the possible models by constraining the model. Roughly speaking, the low values of γ in (2.39) impose a weak constraint and put more weight on the empirical risk minimisation, while large values of the regularisation parameter give more importance to the minimisation of the regularisation function. The regularised empirical risk minimisation problem is a trade-off between fitting the data with the model and constraining the model to stay in a small well-chosen, problem-dependent subset of functions. Furthermore, the balance between satisfying the constraint on the model and staying close to the data is governed by the regularisation parameter.

It is further interesting to notice that Hadamrad initially reported that ill-posed problems were restricted as a mathematical phenomenon and that the real-life problems were "well posed". However, it was latter found that many actual inverse problems are ill-posed. This is true in a large number of fields, from meteorology, hydrology, and mechanics to geophysics or statistics (as we demonstrated with simple example in the previous section). A classical example for linear ill-posed problem is a Fredholm general integral equation of the first kind:

$$\int_a^b K(x,u)\, f(u)du = F(x), \quad a \le x \le b \tag{2.40}$$

where *K* is known squared integral kernel, and *f* is a sought solution.

On the other hand, let us now describe the concept of *Hadamrad well-posedness*. The problem (2.37) is a well-posed if the following conditions hold:

1. $\forall F \in \boldsymbol{F},\ \exists f \in \Lambda, \boldsymbol{A}f = F$. That means a solution to (2.37) exists.
2. $\forall (f_1, f_2) \in \Lambda^2, \boldsymbol{A}f_1 = \boldsymbol{A}f_2 \Rightarrow f_1 = f_2$. The solution is unique.
3. With $\boldsymbol{A}f = F$ and $\boldsymbol{A}f_\delta = F_\delta$, we have $\lim_{F \to F_\delta} f = f_\delta$. The solution is stable with small variations in the right-hand side of (2.37).

The third condition above is equivalent to writing that the inverse operator $\boldsymbol{A}^{-1}$ continuous. One can say that in the context of this study, the inverse operator is the learning procedure. Learning procedures based on the ERM principle (such as minimisation of the quadratic loss function in the regression estimation example) are not stable and the learning problem is thus ill-posed.

The definition of the Hadamard well-posedness does not accommodate a number of tasks such as parameter restoration. This required an extension of the definition of an ill-posed problem. Tikhonov (1963) stated that well-posedness restricts the definition above to a set $\boldsymbol{R} \in \Lambda$. The restriction made by Tikhonov is reflected in the following result: If the operator $\boldsymbol{A}$ is non-ambiguous and continuos on a compact set $\boldsymbol{R}$, then the inverse operator $\boldsymbol{A}^{-1}$ is continuous on the image $\boldsymbol{AR}$. Having a continuous operator with

continuous inverse on compact sets[3], the stability condition is guaranteed. Tikhonov's well-posedness can be assured by the following conditions: problem (2.37) is well-posed if there exist a subset $R \in \Lambda$, such that:

1. It has a solution, $\forall F \in \mathcal{F},\ \exists f \in \Lambda, Af = F$.
2. The solution is unique, $\forall (f_1, f_2) \in \Lambda^2, Af_1 = Af_2 \Rightarrow f_1 = f_2$.
3. For any sequence $f_i \in R$ and $f \in R$, such that $\lim_{i\to\infty} Af_i = Af$, we have $\lim_{i\to\infty} f_i = f$.

The last condition is especially interesting in the context of a learning procedure based on the minimisation of a given loss function *L*. It means that if we have a series of models $f_i$ such that min $L(f_i) \to$ min $L(f)$, then likewise $f_i \to f$. This is precisely what we were missing earlier. Indeed, the law of large numbers does guarantee the convergence of the empirical risk to the expected risk, but *only in the case where the problem is well-posed will the corresponding convergence be true for the solution of the minimisation problem* (White, 1989; Vapnik 1995).

### *2.7.2 Tikhonov regularisation method*

In this section we shell briefly highlight the main concept of the Tikhonov regularisation technique. A main contribution of Tikhonov is that he proposed a method for turning an ill-posed problem into a close well-posed problem. The idea is to continue the learning problem to a restricted set by use of a regularisation functional $\Omega(f)$ (2.39). In the context of the empirical risk minimisation in parametric regression, this functional will typically depend on the model parameters. Setting a constraint on this functional $\Omega(f) \leq c$ defines a structure of subsets of the function set $f \in \Lambda$. Under these conditions, the regularised empirical risk minimisation problems we wish to solve can be written as:

$$\hat{\alpha}_c = \arg\min_{\Omega(f)\leq c} R(\alpha) \tag{2.41}$$

Equation (2.41) is equivalent to seeking the function minimising the empirical risk in a small subset of Λ. The problem here is that it is difficult to carry out minimisation with inequality constraints. However, according to the Kuhn and Tucker theorem (see e.g. Fang, 1993), there is an implicit equivalence between solving (2.41) and minimising modified version of the risk functional:

$$\hat{\alpha}_\delta = \arg\min_{\delta} \left( R(\alpha) + \gamma(\delta)\Omega(f) \right) \tag{2.42}$$

which leads us to the regularisation technique described with equation (2.39). Note that this is reminiscent of the optimisation method of Lagrange multipliers.

[3] The image of a compact set through a continuous operator is compact.

### *2.7.3 Regularisation functionals*

In the previous section we introduced the concept of the regularisation and the use of the regularisation functonal $\Omega(f)$ in order to make the learning problem well-posed. Nonetheless, not all functionals are well suited to be used in regularisation of ill-posed learning problems. As we discussed above the regularisation should actually define a structure of compact sets of functions. In order for the functional $\Omega$ to be suitable, it has to fulfil a number of conditions (Dontchev and Zollezi, 1992):

1. $\Omega$ is semi-continuous in a dense subset of $\Lambda$. This is the case for any continuous function on $\Lambda$.
2. $\Omega$ is positive: $\forall f \in \Lambda, \Omega(f) \geq 0$.
3. A solution of problem (2.37) exists in the domain of definition of $\Omega$.
4. $\Omega$ defines a structure of compact sets: $\forall c \geq 0, \{f | \Omega(f)\} \leq c$ are all compact.

If all these conditions are met, according to Dontchev and Zollezi (1992), $\Omega$ deserves the name of the *regularisation term*. For a regularisation term $\Omega$, the minimisation problem (2.42) is a well-posed problem. The above conditions are far from being restrictive. This allows for a large class of regularisation functinals to be used. In particular, a common choice of $\Omega$ consists in taking a norm on $\Lambda$, $\Omega(f)=\|f\|_p$ or some power of this norm. In the same line, another common choice is to use an operator $L$, typically a derivative operator $\Omega(f) = \|Lf\|_p$.

We would finally like to round off this section describing the impact of the general concept of regularisation with the following citation from Vapnik (1995):

> "…The influence of the philosophy created by the regularisation theory for solving ill-posed problem is very deep. Both the regularisation philosophy and the regularisation techniques become widely disseminated in many areas of science including optimisation, control theory, machine learning and statistics…"

## 2.8 The paradigm of solving learning problems based on the Structural Risk Minimisation principle

The ERM principle described previously, though enriched with the regularisation theory, is intended for dealing with large data sample sizes. In the typical engineering real-life problems one deals with limited amount of data. Clearly, there was a need for a theory, which goes beyond the ERM principle, that is, a theory for controlling the generalisation ability of learning machines or constructing an inductive principle for minimising the risk functional (2.2) using a small sample of training data. This theory was constructed in the late 1960s by Vapnik and Chervonenkis (1968, 1971). The remarkable element of this theory is a collection of different concepts, the so-called *capacity* concept of the learning

machine. Roughly speaking, for a given learning task, with a given finite amount of training data, the best generalisation performance will be achieved if the right balance is found between the accuracy attained on a particular training set, and the *capacity* of the learning machine, that is, the ability of the machine to learn any training set without error (or an error which has a certain *bound*).

Another important concept is the so-called *VC dimension* (Vapnik—Chervonenkis dimension), or more precisely the VC dimension of the set of functions implemented by the learning machine, which is the measure of the notion of capacity mentioned above. It was found that both the necessary and sufficient conditions of consistency and the rate of convergence of the ERM principle depend on the capacity of the set of functions implemented by the learning machine (Vapnik and Chervonenkis, 1989). In particular, it was proven that *distribution-free bounds* on the rate of uniform convergence of the ERM principle depend on the VC dimension, the number of training errors, and the number of observations. This form of bounds led to a new induction principle for controlling the generalisation ability of the learning machines, the so-called Structural Risk Minimisation (SRM) principle[4]. It is thus the SRM principle that opened up new possibilities for inducing (in a real sense) models from and development of new directions in data-driven modelling as sub-symbolic process descriptors. In this section we briefly describe the main concepts of a bound on the generalisation ability of the learning machines and the SRM induction principle. We will also further demonstrate how SRM principle can be linked with ANNs. Finally, we will discuss how the Bayesian approach in learning theory, although has a substantial place in the classical paradigm of function estimation, bringsus to the same scheme and idea as the SRM principle.

### *2.8.1 A bound on the generalisation performance of the learning machine*

In the late 1970s the investigations in the rate of convergence of the learning machines resulted in a family of bounds governing the relation between the capacity of a learning machine and its performance (generalisation ability). The theory explored the considerations under what circumstances, and how quickly, the mean of some empirical quantity (empirical risk) converges uniformly, as the number of available data increases, to the true mean (which would be calculated from an infinite amount of data) (Vapnik, 1979). One of the most remarkable results of the statistical learning theory is the existence of a distribution-free upper-bound on the expected risk $R(\alpha)$ (for a fixed, finite number of observations). Keeping in mind the setting of our learning-from-data problem (see Section 2.2), given a set of i.i.d. observables $\{(\boldsymbol{x}_i, y_i), i=1\ldots N\}$ generated according to an unknown probability density, and our learning machine (set of functions $f(\boldsymbol{x}, \alpha)$) with

[4] See monograph by V.N.Vapnik: *Estimation of Dependencies Based on Empirical Data*, Nauka, Moskow, 1979. English translation: Springer-Verlag, New York, 1982.

the task to learn the mapping $\boldsymbol{x}_i \rightarrow y_i$, the following bound on the risk holds (Vapnik, 1995) with probability $1-\eta$:

$$R(\alpha) \le R_{emp}(\alpha) + \sqrt{\frac{h\left(\log(2N/h)+1\right)-\log(\eta/4)}{N}}\ , \qquad (2.43)$$

where $h$ is non-negative integer called VC dimension of this set of functions and is a measure of the notion of the capacity of the learning machine and $N$ is the number of the training data patterns. The right-hand side of the equation (2.43) is usually called the *bound on the risk*. Some authors (Guyon et al., 1992) call it the "guaranteed risk", but this is arguable since it is really bound on the expected risk, not a risk, and it holds only with a certain probability, and therefore is not guaranteed. The second term on the right-hand side is known as the *VC confidence*.

We would like to stress here three points about this bound on the risk (generalisation error). First, remarkably, it is independent of any probability density $p(\boldsymbol{x}, y)$. It assumes only that both the training and the test data are drawn independently according to some probability density $p(\boldsymbol{x}, y)$ (see Section 2.2). Second, it is usually not possible to compute the left-hand side of the equation (2.43) directly. Finally, if we know the VC dimension $h$, we can easily compute the right-hand side of the equation (2.43). Thus, given (or properly chosen) several learning machines (several sets of functions), and choosing a fixed, sufficiently small $\eta$, by then taking the machine which minimises the right-hand side of (2.43) gives the minimum lowest upper bound on the actual risk. This gives an inductive principle for choosing a learning machine for a given learning task, and is the essential idea of the *structural risk minimisation principle*. Given a fixed family of learning machines to choose from, to the extent that the bound is tight for at least one of the machines, one will not be able to do better than this. If the bound is not too tight for any of the learning machines, the hope is that the right-hand side still gives useful information (satisfactory accuracy and generalisation ability) as to which learning machine minimises the actual risk.

### *2.8.2 Structural Risk Minimisation (SRM) principle*

We briefly summarise here the principle of structural risk minimisation (SRM) (Vapnik, 1979). We mentioned earlier that the ERM inductive principle can deal with large data sample sizes. Considering the inequality (2.43), when the ratio $N/h$ (ratio between the number of training samples to the VC dimension) is large, the VC confidence becomes small. The actual risk is then close to the value of the empirical risk. In this case, a small value of the empirical risk $R_{emp}(\alpha)$ guarantees a small value of the expected risk $R(\alpha)$. However, when the ratio $N/h$ is small (limited amount of data to learn from), a small $R_{emp}(\alpha)$ does not guarantee a small value of the risk $R(\alpha)$. In this case, in order to minimise the risk $R(\alpha)$, one has to minimise the right-hand side of (2.43) simultaneously over both terms: the empirical risk and the VC confidence.

Note that the VC confidence term depends on the chosen class of function, whereas the empirical risk and the actual risk depend on one particular function chosen by the learning procedure. Our goal is to find a particular subset of the chosen set of functions,

such that the risk bound for that subset is minimised. Clearly we cannot arrange that the VC dimension as a controlling variable in the optimisation procedure will vary smoothly, since it is an integer. Instead one can introduce a "structure" by dividing the entire class of functions into nested subsets (see Figure 2.9).

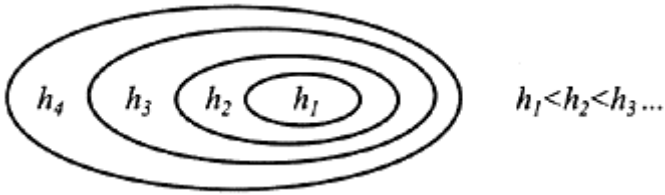



Figure 2.9. Nested subset of functions ordered by VC dimension.

For each subset, one must be able either to compute *h*, or to get a bound on *h* itself. SRM then consists of finding the subset of functions, which minimises the bound on the actual risk. Simply training a set of machines, one for each subset can do this, where for a given subset the goal for training is to minimise the empirical risk. One then takes that trained machine in the series whose sum of empirical risk and VC confidence is minimum (optimal), Figure 2.10.

The general SRM principle can be implemented in many ways. For example, there are several possible ways to implement the SRM principle for a set of functions used by ANNs:

1. For a fully connected feed-forward neural network (Figure 2. 8) in which the number of units in one of the hidden layers is monotonically increased, the set of implementable functions define a structure as a number of hidden units is increased. The risk on this structure can be further minimised.
2. Consider a set of functions S={$f$= ($x$, $\theta$), $\theta \in W$ }, implementable by an ANN (learning machine) with fixed architecture, where the parameters **{θ}** are the weights of the neural network. A structure can be introduced through $S_p$={$f(x, \boldsymbol{\theta})$, $\|\boldsymbol{\theta}\| \le c_p$} and $c_1 < c_2 < \ldots < c_n$. Under the general loss function, the minimisation of the empirical risk within one element $S_p$ of the structure introduced can be done by minimizing the functional

$$R_{cmp}(\boldsymbol{\theta}, \gamma_p) = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta})\right) + \gamma_p \|\boldsymbol{\theta}\|^2 \quad ,$$

which with appropriately chosen Lagrange multipliers leads us to the well-known *weight decay* estimation procedure (Plaut et al., 1986; Krogh and Hertz, 1992).

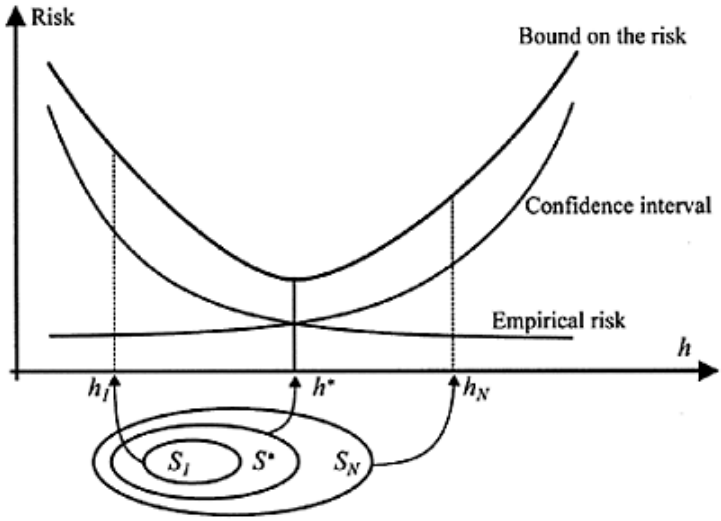



Figure 2.10. The bound on the risk is the sum of the empirical risk and the VC confidence. The smallest bound on the risk is achieved by taking particular trained machine on an appropriate subset of the structure whose sum of the empirical risk and the VC confidence is minimal (adopted from Vapnik, 1998).

3. The structure can be imposed on the input representation to the ANN with fixed architecture. The input can be modified by introducing a transformation $\boldsymbol{z}=K(\boldsymbol{x},\beta)$, where the parameter β controls the type of the transformation (e.g. the width of the smoothing kernel). A structure can be introduced in a set of functions $S=\{f(K(x,\beta),\theta), \theta \in W\}$ through $\beta \geq c_p$ and $c_1>c_2>\ldots> c_n$. The SRM principle can be then implemented by estimating the VC dimension (confidence) for each of the elements $S$ of the structure and by minimising the empirical risk. In this way the upper bound on the actual risk can be minimised.

The structural risk minimisation inductive principle has laid the ground for the emerging computation learning technique known as Support Vector Machine (SVM).

### *2.8.3 Support vector machine*

Support vector machine is relatively new computational learning technique, which embodies the SRM principle. The main idea of the support vector machine is to map the input vector x into a high-dimensional feature space Z by using nonlinear mapping

(kernel functions) chosen *a priori*, and then an optimal separating hyperplane is constructed using the SRM inductive principle. The optimal separating hyperplane passes through the given points (patterns) from the training data set, which are found by solving dual quadratic optimisation problem. The points (vectors) are termed *support vectors*.

A detailed description of the concept and the algorithm of SVM is outside of the scope of this thesis. For introduction to the subject of support vector machines and the SRM principle we refer to Vapnik (1995, 1998), Burges (1998), Saunders et al. (1998), Schölkopf (1997) and Smola (1996). A brief description of SVMs for pattern recognition and their application in a framework of hybrid data-driven model is given in Chapter 6.

Since their introduction (Vapnik, 1995), SVMs have attracted the attention of the researchers and practitioners due to their solid theoretical background, based on the statistical learning theory and the SRM principle, and their increasing successful application to real-life problems in both the pattern recognition and regression estimations. For the pattern recognition case, SVMs have been used for hand-written digit recognition (Cortes and Vapnik, 1995; Burges and Vapnik, 1995; Scholkopf, Burges and Vapnik, 1996) object recognition (Blanz et al., 1996), voice identification (Schmidth, 1996), face image detection (Osuna et. al., 1997) and text categorisation (Joachims, 1997). For the regression estimation case SVMs have been compared on benchmark time series prediction test (Muller et al., 1997; Mukherjee et al., 1997) on artificial data (Vapnik, Golowich and Smola, 1996) and for dynamic reconstruction of the well-known Lorenz chaotic system (Mattera and Haukin, 1999). Dibike, Velickov and Solomatine (2000a and 2000b) with Babovic and Kajzer (2000) have pioneered the application of SVMs for solving civil engineering problems. Velickov et al. (2000), have demonstrated and compared SVMs with other sub-symbolic model induction engines for automated land cover classification of remote sensed images for the purposed of the hydrological modelling. A novel hybrid algorithm was also developed and reported.

### *2.8.3 Bayesian learning paradigm and the SRM principle*

Tomas Bayes was a British cleric and amateur mathematician (it appeared a very good one), who died in 1691. Among his papers was found a curious unpublished manuscript, which was then published in 1763 (see Molina, 1963 for a photographic reproduction of the work and some historical comments) and gave rise to a new learning paradigm, termed with different names, such as “Bayesian learning” or “Bayesian approach” or “Bayesian statistics”. Latter on, in almost his first published work (1794), Laplace rediscovered Bayes’ principle in greater clarity and generality, and then for the next 40 years proceeded to apply it to various problems of astronomy, geodesy, meteorology and statistics. The Bayesian learning paradigm is founded upon the premise that all forms of uncertainty can be expressed and measured by probabilities (Bernardo and Smith, 1994). Although the paradigm can be expressed in a formal framework, based on mathematical abstraction and rigorous analysis, it relies upon subjective experience. That is, it offers a rationalist and coherent theory where all kinds of uncertainties (e.g. parameters of the model, models, process uncertainties) are described in terms of subjective beliefs or probabilities. However, once the individual beliefs of uncertainties are expressed, and assuming access to the same data, the results should be unique and reproducible.

Bayesian learning paradigm is based on the rather simple chain rule (or theorem) known as Bayes' rule, yet it is by far one of the most important principles underlying the scientific inference (see Jaynes, 1995). Simple application of the conditional probability definition allows us to derive the Bayes rule. Denoting two events (or proposition) by *A* and *B*, and applying the basic product and sum rules of probability we have:

$P(A \cap B)=P(A|B)P(B)$

$P(B \cap A)=P(B|A)P(A)$ (2.44)

$$P(A \mid B) + P(\overline{A} \mid B) = 1 \tag{2.45}$$

As we obviously have $P(A \cap B)=P(B \cap A)$ and if $P(B)>0$, we get from (2.44) the well-known Bayes' rule (although Bayes never wrote it):

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{2.46}$$

It can be also shown that in (2.46) an additional event/proposition *C* can be introduced, such that:

$$P(A \mid BC) = P(A \mid C)\frac{P(B \mid AC)}{P(B \mid C)} \tag{2.47}$$

One can ask the question: But what is so important in (2.47) apart from it being just a statement that the product probability rule is consistent? The important thing is that in the Bayes' rule (2.27) we have a *mathematical representation of the process of learning;* exactly what we need for our extended logic that allows induction of models from data. $P(A|C)$ is our *prior probability* for *A* when we know only *C*. $P(A|BC)$ is its *posterior probability,* updated as a result of acquiring new information *B*. Typically and very generally, *A* represents some hypothesis or theory (or model), whose truth we wish to ascertain, *B* represents the new data from observations, and *C* represents the background information (knowledge), that is, the totality of what we knew (and believed) about *A* before getting the data *B*. The other distributions on the right-hand side are the *likelihood* and the *evidence* (also known as innovation or predictive distribution). Thus the Bayes rule can be written in a form:

$$posterior = prior\,\frac{likelihood}{evidence} \tag{2.48}$$

Our subjective beliefs and views on the uncertainty are expressed in the prior distributons. Once the data is available the evidence allows us to update these beliefs. The resulting posterior distribution incorporates both our *a priori* knowledge and the information conveyed by the data, and thus improves on our common sense. In the first place it is clear that the prior probability $P(A|C)$ is necessarily present in all inductive inference; therefore to ask the question of type "What do you know about *A* after seeing *B?*" cannot have any definitive answer, because is not a well-posed question in the Bayesian learning paradigm, if we fail to take into account the question "What did you

know about *A* before seeing *B?*". This reasoning is crucial to judge the frequently repeated phrase "Let the data speak for themselves!". Our view on this phrase is: *they cannot and never have*. For example, if we want to decide between various learning machines (data models) but refuse to supplement the data with prior information (incorporating our background knowledge and understanding of the relationships between physical processes being modelled) about them, any probabilistic inference will lead us to favour the "Sure Thing" (ST) model, according to which, for example, every millisecond of detail of the dynamical system was inevitable; nothing else could have happened. For the data we will always have much higher probability (close to 1) on the ST model than on any other model. Only by supplying proper prior information could the ST model be rejected.

The other remarkable think is that Bayes' rule also allows to produce several levels of inference, as probabilities conditioned on C can be in turn calculated using Bayes' rule. If we write for example that $P(A|BC)=P(A|C)(P(B|AC)/P(B|C))$, we can in turn combine this with the result of another inference in the form $P(A|C)=P(A)(P(C|A)/P(C))$. This formalism is very useful to incorporate new knowledge in our inference, or to update the results (model) once new information is available. In this way we can apply Bayes' rule repeatedly as new pieces of information $B_1$, $B_2$,…arrive, thus the posterior probability from each application is becoming the prior probability for the next. This raises a possibility for effective sequential learning (training of the machine) since at any stage the probability that Bayes' rule assigns to *A* depends only on the total evidence $B_{tot}=B_1$, $B_2$,…, $B_N$. One can reach the same learning performance by a single application of the Bayes' rule using $B_{tot}$.

Although in the classical paradigm of learning from data and especially function estimation an important place belongs to the Bayesian approach (see Berger, 1985 for overview), in the last decade Bayesian learning paradigm is receiving increasing attention (Berger, 1999) due to the reasons briefly discussed above. The Bayesian learning paradigm and especially the investigation of the ability to induct models describing dynamical systems from data (dynamic Bayesian networks) is one of the major focuses of this thesis (see Chapter 4).

However, one of reasons we introduced the Bayesian learning paradigm in this section was the intention to show that Bayesian learning implicitly embodies the SRM induction principle. Consider, for simplicity, the problem of regression estimation from measurements corrupted with independent additive noise (same as in section 2.3.3)

$$y_i=f(x_i, \alpha_0)+\varepsilon_i. \tag{2.49}$$

In order to estimate the regression model in within the ML framework, one had to know a parametric set of functions $f(\mathbf{x}, \alpha)$, $\alpha \in \Lambda \subset R^n$, that contain the regression function $f(\mathbf{x}, \alpha_0)$, and to know (or assume) the noise distribution model $p(\varepsilon)$.

In the Bayesian approach, additional information needs to be supplied: one has to know the *a priori* density function $p(\alpha)$ that for any function from the parametric set of functions $f(\mathbf{x}, \alpha)$, $\alpha \in \Lambda$ defines the probability for it to be the appropriate model. If $f(\mathbf{x}, \alpha_0)$ is the regression model, then the probability of the training data $D=(x_1, y_1),\ldots, (x_N, y_N)$ can be written as:

$$P(D \mid \alpha_0) = \prod_{i=1}^{N} P\left(y_i - f(x_i, \alpha_0)\right) \tag{2.50}$$

Having seen the data, the *a posteriori* probability that the parameter α defines the regression model can be estimated applying the Bayes' rule:

$$P(\alpha \mid D) = \frac{P(\alpha)\, P(D \mid \alpha)}{P(D)} \tag{2.51}$$

This expression can be used to choose the approximation to the regression model, using different estimation frameworks (such as maximum *a posteriori* or evidence framework, see Chapter 4 for details). Choosing the approximation function $f(x, \hat{\alpha})$ that maximises the conditional probability (2.50) on $\hat{\alpha}$ is equivalent to the following functional:

$$\Phi(\alpha) = \sum_{1}^{N} \ln P\left(y_i - f(x_i, \alpha)\right) + \ln P(\alpha) \tag{2.52}$$

If we simply consider that the noise distribution is according to the normal law,

$$p(\varepsilon) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right) \tag{2.53}$$

the functional (2.53) can be written as:

$$\Phi^{*}(\alpha) = \frac{1}{N} \sum_{1}^{N} \left(y_i - f(x_i, \alpha)\right)^2 - \frac{2\sigma^2}{N} \ln P(\alpha) \tag{2.54}$$

which has to be minimised with respect to the parameter a in order to find the approximation function $f(x, \hat{\alpha})$. The first term of the functional (2.54) is in fact the empirical risk, and the second term can be regarded as a regularisation term. Therefore the Bayesian approach brings us to the same scheme that is used in the SRM inductive principle.

To summarise, in order to use the Bayesian learning framework, one must posses the following strong *a priori* information:

(i) The given set of functions of the learning machine should coincide with the type of the engineering problems to be solved.

(ii) The *a priori* distributions on a particular modelling problem (or the various uncertainties) are expressed by subjective beliefs using the domain information and knowledge.

One can argue that these two requirements are positive or negative requirements in the process of model induction from data. Some authors (see Vapnik, 1995) argue that due to the human-machine type of inference this is the only shortcoming of the Bayesian approach (since machine learning in a real sense should induct the models from data),

whether others (see Hecerman, 1996) elaborate that these requirements are one of the major advantages of the Bayesian learning framework. This debate is in fact the essence of this thesis. We strongly argue that for engineering problem-solving and modelling purposes (e.g. hydrodynamic modelling, hydrological modelling, environmental modelling, etc.) the process of model induction from data is an *interactive* process *mobilising* as much as possible of the available knowledge of the underlying physical processes and already known relationships between various variables describing them. These kinds of prior domain information/knowledge should guide the process of learning models from data, which in turn we hope to improve our knowledge and beliefs about the factual physical systems and processes (and sometimes lead to new discoveries) throughout the perception and cognition of the induced models (viewed in terms of the definition of model described in Chapter 1).

The first requirement of the Bayesian learning framework mentioned above can be interpreted as the required regularisation (see Section 2.7) in order to avoid solving ill-posed problems by imposing certain constraints on the set of functions (learning machine). For example, in a case of hydrodynamic modelling problem, one could express those constrains by the first principles for Hamiltonian type of systems translated as conservation of mass, momentum, and energy.

The second requirement of the Bayesian learning paradigm is probably at the heart of data-driven modelling. The real Bayesian approach advocates that one should express beliefs about various kinds of uncertainties (e.g. model parameters, type of the models, inherent process uncertainties, uncertainties due to a lack of knowledge about specific processes) present in the modelling problem without necessarily seeing the data beforehand. This in turn might be a difficult task for analysis of data that are observed on some new phenomenon and processes. The *data exploratory analysis* should play an important role in this case, hereby providing a way of using the data to assist in specifying the prior information to the learning machine. This approach is sometimes termed as a empirical Bayesian framework, and is being frequently used in practice.

The fact that the Bayesian learning paradigm can also be used for machine learning (in a true sense) is due to the reason that Bayesian framework inherits the notions of regularisation and capacity control, as we discussed above. The advocates of the pure machine learning approach (machine-only type of inference) favour weak *a priori* qualitative information about the reality being modelled. Our view is that these kinds of approaches can be of use only in the analysis of huge amounts of data and information, which in general do not necessarily describe some (dynamical) physical processes. Thus they can be seen rather as a technique, but not as a real modelling tool. This discussion provokes us to raise the same question, which was raised many times since the invention of the computer: The question about the existence of a “real thinking computer”. Am interesting answer to this question was given by J.von Neumann (quoted by Jaynes, 1995):

> “…You insist that there is something that a machine can do. If you will tell me precisely what it is that a machine cannot do, then I can always make a machine which will do exactly just that!…”

In principle, the only operations which a learning machine cannot perform for us are those we cannot describe with sufficient details and knowledge, or which cannot be completed in a finite number of steps. Of course, to argue about it we need to point to the existence of human brain, which in fact *does it*. Just as in the citation above, the only limitations on making a machine which learns from data and tries to perform useful inductive reasoning are our own limitations in not knowing exactly what "learning from data" consists of.

## 2.9 Search for the minimum of risk in solving learning problems

We have introduced earlier in this chapter that a significant part of the problem of model induction from data, using the ERM and SRM inductive principles or Bayesian framework, consists of searching for some kind of a minimum of the actual risk (either expressed through the empirical risk or the smallest upper bound on the expected risk). This in turn requires employment of some optimisation technique. The purpose of this section is to give a brief description of the optimisation techniques, which in one way or another are used throughout this work. Optimisation methods are covered in many books such as e.g. (Fletcher, 1987) or (Press et al., 1992) for one-dimensional and multi-dimensional problems. For comparative analysis of different global optimisation methods and search strategies in a relation to model calibration refer to Solomatine (2000). Some of the optimisation methods discussed here (such as quadratic approximation and conjugate gradient) are necessary to handle nonlinear problems. However, most of the methods presented in this section are also used in the linear case. Indeed, iterative minimisation methods are a common alternative to the computationally expensive matrix inversion methods.

### *2.9.1 Back-propagation method*

In section 2.6.3 we introduced the artificial neural networks as nonlinear nonparamentric learning machines, where the so-called *back-propagation method was* one of the important achievements in the general learning theory. The back-propagation method is usually accredited to LeCun (1986) or Rumelhart et al. (1986). However, it was discovered earlier in different contexts. Vapnik (1995) mentions its use in Bryson et al., (1963) for solving some control problems. Bottou (1991) cites Amari (1967) in the context of adaptive systems and notes that it is nothing more than a proper application of the derivation rules invented by Leibnitz in the $17^{th}$ century. This method is indeed an application of chain rule of derivation to the MLP type of ANNs. We present here the derivation for a one input, one-hidden layer case, but the generalisation is straightforward.

The local quadratic loss function (sometimes termed as cost) for data example $k$ is expressed as:

$$L_k(\theta)=(y_i-y(x_k))^2 \tag{2.55}$$

For a simplistic calculation we introduce the output $h_j$ of a hidden unit $j$, and its input as $p_j = h^{-1}(h_j)$. The derivatives of the loss function with respect to the input of the hidden layer and the input layer are:

$$\frac{\partial L_k}{\partial p_j} = \frac{\partial h_j}{\partial p_j}\frac{\partial y}{\partial h_j}\frac{\partial L_k}{\partial y} = 2h'(p_j)\Theta_j\left(y_k - y(x_k)\right) \tag{2.56}$$

$$\frac{\partial L_k}{\partial x_i} = \sum_{j=1}^{H}\frac{\partial p_j}{\partial x_i}\frac{\partial L_k}{\partial p_j} = \sum_{j=1}^{H}\theta_{ji}\frac{\partial L_k}{\partial p_j} \tag{2.57}$$

The derivatives with respect to the parameters are:

$$\frac{\partial L_k}{\partial \Theta_j} = \frac{\partial y}{\partial \Theta_j}\frac{\partial L_k}{\partial y} = 2h_j\left(y_k - y(x_k)\right) \tag{2.58}$$

$$\frac{\partial L_k}{\partial \theta_{ji}} = \frac{\partial p_j}{\partial \theta_{ji}}\frac{\partial L_k}{\partial p_j} = x_i\frac{\partial L_k}{\partial p_j} \tag{2.59}$$

The calculation of the neural network estimation is done by a forward pass, introducing $h_j$ and $y(x)$ given the $x_i$, $\theta_{ji}$ and $\Theta_j$. The calculation of the derivatives of the loss with respect to the parameters is done by back-propagation, using (2.56) and (2.57), then (2.58) and (2.59). The first order derivatives with respect to the parameters allow the use of first order optimisation method to minimise $L(\theta)$ (and even some approximations of second order methods). The first order derivatives with respect to the inputs allow the use of a neural network as a part in a *modular* or *hybrid model* as presented latter.

### *2.9.2 Quadratic optimisation*

For one-dimensional optimisation, the basic techniques are for example golden search or parabolic interpolation. These are not directly relevant to the machine learning, though they are crucial when performing a *line search*, i.e. minimising along a given search direction. The quadratic optimisation is also relevant for to the minimisation in a multi-dimensional space (Fletcher, 1987). Given a cost function $L(\theta)$, one can perform a quadratic expansion around $\hat{\theta}$, such as:

$$L(\theta) = L(\hat{\theta}) + \frac{1}{2}\left(\theta-\hat{\theta}\right)^{T}\mathbf{H}\left(\theta-\hat{\theta}\right) \tag{2.60}$$

where **H** is the Hessian matrix (matrix containing second order partial derivatives) of the loss function. This expression relates to the use of a *quadratic loss* function. It is exact for linear models and gives a good approximation to the nonlinear models.

### *2.9.3 Steepest descent*

The steepest descent algorithm dates back to Cauchy (see Press et al., 1992). The main idea of this algorithm is at every iteration to search along the direction of the steepest descent i.e. the gradient:

$$\theta_{t+1} = \theta_t - \eta_t \nabla L(\theta_t) \tag{2.61}$$

where $\eta_t$ is a gain parameter that can be initiated in different ways. For example:

- setting $\eta_t$ to a constant parameter;
- finding the "optimal" rate of change: $\hat{\eta}_t = \arg\min_{\eta} L\left(\theta_t - \eta_t \nabla L(\theta_t)\right)$.

The rate of convergence of an algorithm can be defined as a speed at which it gets closer to the solution, or in other words, the ratio between the distance to the solution at time *t* and the distance to the solution at time *t-1*. Its convergence rate is a standard topic of improvement in numerical analysis (e.g. Møller, 1993), due to the well-known reason that when the loss surface has a very narrow valley around the optimal solution $\hat{\theta}$, the algorithm jumps from one side to the other of the valley in very short orthogonal steps, which in turn results in a large number of iterations. Several improvements have been suggested such as adding a momentum term (see Møller, 1993).

### *2.9.4 Stochastic gradient descent*

The stochastic version of the gradient descent algorithm, sometimes referred as on-line gradient descent, can lead to a drastic improvement in the convergence. The application of the algorithm and the convergence analysis date back to 1950s. Rumelhart et al. (1986) introduced and applied this algorithm for the first time to ANNs. The stochastic algorithm consists of updating the parameters on the basis of the gradient of the *local* loss:

$$\theta_{t+1} = \theta_t - \eta_{t+1} \nabla L_k(\theta_t) \tag{2.62}$$

where the $L_k$ is the local loss for a pattern *k:* $\|y_k - f_{\theta t}(xk)\|^2$. The choice of the pattern *k* to use in time step *t* is done in a stochastic manner, taking the training examples at random. This updating scheme is well suited for on-line adaptation of the parameters, while the previous algorithm uses the computation of the gradient on the entire training set, making it better suited for off-line (or batch) training. The stochastic gradient descent is recommended for learning problems such as higher dimensional pattern recognition, where the training set is large, and often redundant. It is generally less well suited to small scale pattern recognition and regression estimation. In such cases, the conjugate gradient algorithm presented below usually yields better results. For a good introduction and demonstration of the convergence of the algorithm we refer to Bottou, (1991).

### *2.9.5 Conjugate gradient*

The conjugate gradient method increases the efficiency of the optimisation by avoiding the oscillations of the steepest descent method in ill-posed cases. This algorithm is being studied extensively in the linear optimisation literature. A neural network perspective is described in Møller (1993). A brief description of the conjugate gradient is presented below.

Let us first recall that two vectors $u$ and v are said to be *conjugate* with respect to matrix $\mathbf{A}$ if: $u^{\mathrm{T}}\mathbf{A}v=v^{\mathrm{T}}\mathbf{A}u=0$, which means that they are orthogonal in the sense of the quadratic form. As we discussed earlier, the quadratic approximation (2.60) involves the Hessian matrix of the loss function. Furthermore, during the search it would be convenient that once a minimisation step is completed, the next search direction does not interfere with the previous one (in order to avoid reversing the previous gain). In other words, it would be good if the choice of the next search direction is restricted to the conjugate hyperplane, that is, the old and the new search directions are kept conjugate with respect to the Hessian matrix $\mathbf{H}$. One can construct a set $h_t$ of successive conjugate search directions together with a set $g_t$ of gradient directions. If $\theta_t$ is the approximation of the location of the minimum of $L(\theta)$ at step $t$, then the next gradient directions can be expressed as:

$$g_{t+1} = \nabla L(\theta_t) = \mathbf{H}\left(\theta_t - \hat{\theta}\right). \tag{2.63}$$

The next search direction can be taken as a combination of the gradient direction and the previous search direction:

$$h_{t+1}=g_{t+1}-\gamma_{t+1}h_t, \tag{2.64}$$

where $\gamma_{t+1}$ is a scalar. We recall that the $h_i$ are a conjugate family of vectors, and in particular $h_{t+1}{}^{\mathrm{T}}\mathbf{H}h_t=h_{t-1}{}^{\mathrm{T}}\mathbf{H}h_t=0$. This property is used together with (2.64) at step t+1 and to obtain the value of the $\gamma_{t+1}$:

$0=g_{t+1}{}^{\mathrm{T}}\mathbf{H}h_t-\gamma_{t+1}h_t{}^{\mathrm{T}}\mathbf{H}h_t$ and $h_t{}^{\mathrm{T}}\mathbf{H}h_t=gt^{\mathrm{T}}\mathbf{H}h_t$,

which results to:

$$\gamma_{t+1} = \frac{g_{t+1}{}^{\mathrm{T}}\mathbf{H}h_t}{g_t{}^{\mathrm{T}}\mathbf{H}h_t}. \tag{2.65}$$

This expression still involves several computations of the Hessian matrix, the calculations of which one in general wants to avoid. By noticing the property that as θt is obtained by a search along the direction $h_t$ starting from $\theta_{t-1,}$ there exists a value $a_t$ such that $\theta_t-\theta_{t-1}=a_th_t$. From (2.63) one can write:

$$g_{t+1}-g_t=\mathbf{H}(\theta_t-\theta_{t-1})=a_t\mathbf{H}h_t, \tag{2.66}$$

which leads to a more convenient expression for the $\gamma_{t+1}$:

$$\gamma_{t+1} = \frac{g_{t+1}^{\mathrm{T}}(g_{t+1} - g_t)}{g_t^{\mathrm{T}}(g_{t+1} - g_t)} . \tag{2.67}$$

Yet another consequence of the line search along $h_t$ is that the gradient of the loss function $L$ in $\theta_t$ is orthogonal to the search direction, $h_t^{\mathrm{T}} g_{t+1} = 0$. Using (2.66) and (2.64), this can be extended to:

$$h_i^{\mathrm{T}} g_j = g_i^{\mathrm{T}} g_j = 0, \ \forall i \neq j \ . \tag{2.68}$$

In this manner several authors have proposed different versions of the equation (2.67):

$$\gamma_{t+1} = -\frac{g_{t+1}^{\mathrm{T}} g_{t+1}}{g_t^{\mathrm{T}} g_t} \qquad \text{Fletcher-Reeves} \tag{2.69}$$

$$\gamma_{t+1} = -\frac{g_{t+1}^{\mathrm{T}}(g_{t+1} - g_t)}{g_t^{\mathrm{T}} g_t} \qquad \text{Polak-Ribière} \tag{2.70}$$

$$\gamma_{t+1} = -\frac{g_{t+1}^{\mathrm{T}}(g_{t+1} - g_t)}{h_t^{\mathrm{T}} g_t} \qquad \text{Hestenes-Stiefel} \tag{2.71}$$

The effects of the three expressions are indeed the same when the loss function is exactly quadratic. However, in the case of nonlinear regression estimation, this is not the case and the Hessian is not constant. In this case, the last two expressions (2.79) and (2.71) show better performances since they restart the algorithm by resetting the search direction to the gradient of the loss function. Indeed, when two successive gradient directions are too close, then $g_{t+1} \approx g_t$, leading to $h_{t+1} \approx g_{t+1}$. Finally, it should be noted that the conjugate gradient algorithm does not depend on the setting of an extra parameter such as the learning rate in both steepest descent and stochastic gradient descent.

### *2.9.6 Newton and quasi-Newton*

Let us first recall that the derivative in $\theta_t$ of the loss function is: $\nabla L(\theta_t) = \mathbf{H}\left(\theta_t - \hat{\theta}\right)$, which leads to a direct estimation of the minimum:

$$\hat{\theta} = \theta_t - \mathbf{H}^{-1} \nabla L(\theta_t) \ . \tag{2.72}$$

This is similar to the Newton method for finding roots in one-dimensional problems. Equation (2.72) is valid only for exact quadratic loss function. In the general case, the Newton algorithm in multi-dimensional space consists of choosing the search direction according to:

$$\theta_{t+1} = \theta_t - \eta_{t+1} \mathbf{H}^{-1} \nabla L(\theta_t) \ , \tag{2.73}$$

where $\eta_{t+1}$ is set by simple line search. This second order method expressed in (2.73) requires the calculation of the Hessian of the loss function, which one wants to avoid due to high computational demands and the sensitivity of the second order derivatives in the context of nonlinear models. Usually, the quasi-Newton method is used to approximate the Hessian. If one rewrites the quadratic loss function as $L(\theta) = 1/N \sum_{i=1}^{N} \varepsilon_i(\theta)^2$, the Hessian becomes:

$$\mathbf{H}_L(\theta) = \frac{2}{N}\sum_{i=1}^{N}\left(\varepsilon_i(\theta)\mathbf{H}_\varepsilon(\theta) + \nabla\varepsilon_i(\theta)\nabla\varepsilon_i(\theta)^{\mathrm{T}}\right). \qquad (2.74)$$

Neglecting the first term in (2.74) gives the *Gauss-Newton* approximation (Battiti, 1992), where the Hessian can be written as:

$$\mathbf{H}_L(\theta) = \frac{2}{N}\sum_{i=1}^{N}\nabla\varepsilon_i(\theta)\nabla\varepsilon_i(\theta)^{\mathrm{T}} \quad . \qquad (2.75)$$

Since the choice of the proper search direction depends on the inverse of the Hessian (2.73), one can approximate this inverse using the Sherman-Morrison inversion identity by the following iterative formula:

$$\mathbf{H}_i^{-1} = \mathbf{H}_{i-1}^{-1} - \frac{\mathbf{H}_{i-1}^{-1}\nabla\varepsilon_i(\theta)\nabla\varepsilon_i(\theta)^{\mathrm{T}}\mathbf{H}_{i-1}^{-1}}{1+\nabla\varepsilon_i(\theta)^{\mathrm{T}}\mathbf{H}_{i-1}^{-1}\nabla\varepsilon_i(\theta)} \quad i = 1\ldots N \qquad (2.76)$$

where after N iterations, $\mathbf{H}_N^{-1}$ approximates the inverse of the approximate Hessian.

Another method for computing the approximate inverse Hessian, commonly used in practice, is the one-step Broyden-Fletcher-Goldfrd-Shanno (BFGS) method (see for example Battiti, 1992). If one introduces the following differences $u_t = \nabla L(\theta_{t+1} - \theta_t)$ and $s_t=(\theta_{t+1}-\theta_t)$, then the positive definite update for the inverse Hessian is:

$$\mathbf{H}_{t+1}^{-1} = \mathbf{H}_t^{-1} + \frac{s_t s_t^{\mathrm{T}}}{u_t^{\mathrm{T}} s_t}\left(1 + \frac{u_t^{\mathrm{T}}\mathbf{H}_t^{-1}u_t}{u_t^{\mathrm{T}} s_t}\right) - \frac{\mathbf{H}_t^{-1}u_t s_t^{\mathrm{T}} + s_t u_t^{\mathrm{T}}\mathbf{H}_t^{-1}}{u_t^{\mathrm{T}} s_t} \quad . \qquad (2.77)$$

This expression ensures that the Hessian and its inverse are symmetric positive definite. It can be shown that for an exact line search, BFGS is equivalent to the conjugate gradient method.

## 2.10 Summary

The essence of the data-driven modelling, which is closely related to modelling nonlinear dynamical systems based on chaos theory is learning models from data. This learning problem is an ill-posed problem and related to computational intelligence techniques based on search and optimisation. In this chapter we provided a brief review of the

learning theory from statistical and machine learning perspectives and the associated methods and techniques, which are relevant and further used in this work. Both, the classical approaches based on Empirical Risk Minimisation (ERM) principle and the approaches based on Structural Risk Minimisation (SRM) principle especially related to the multivariate regression estimation were addressed. We contributed with an original discussion and demonstrated the parallels between the two approaches. Furthermore, the Bayesian learning paradigm that combines learning from data and prior domain information/ knowledge was introduced and discussed. Finally, our standpoint is that for engineering problem-solving and modelling purposes (e.g. hydrodynamic modelling, hydrological modelling, environmental modelling, etc.) the process of learning models from data is an interactive process mobilising as much as possible of the available knowledge of the underlying physical processes and already known relationships between various variables describing them.

# Chapter 3
# Nonlinear Dynamical Systems and Deterministic Chaos

## 3.1 Introduction

The paradigm of nonlinear dynamics and the concept of deterministic chaos in the last decade have influenced the thinking and problem solving in many fields of science and engineering. As models, chaotic dynamical systems show rich and even surprising variety of dynamical structures and solutions. Most appealing for researches and practitioners is the fact that the deterministic chaos provides a prominent explanation for irregular behaviour and instabilities in dynamical systems, which are deterministic in nature. The most direct link between the concept of deterministic chaos and the real world is the analysis of data (time series) from real systems in terms of the theory of nonlinear dynamics, which is the major focus of this thesis. We consider a "system" in this context as a natural phenomenon or processes, a laboratory experiment, or a numerical simulation.

Linear methods interpret all regular structure in a data set, such as dominant frequency, as linear correlations. This means that the intrinsic dynamics of the system are governed by the linear paradigm that small causes lead to small effects. Since linear equations describing dynamical system can only lead to exponentially growing or periodically oscillating solutions (dynamical evolution of the system), all irregular behaviour of the system has to be attributed to some random external input to the system. On the other hand, as we will demonstrate in this chapter, random input is not the only possible source of irregularity in a system's output: nonlinear dynamical systems can produce very irregular data with purely deterministic equations of motions, caused by slight changes in some of the control parameters and sensitivity to the initial (and/or boundary) conditions. Of course, the systems which exhibit both nonlinearity and random input will most likely produce irregular data as well.

On the other hand, the theory of nonlinear dynamics and data analysis have progressed to the stage where most fundamental properties of nonlinear dynamical systems have been observed in the laboratory and proven theoretically on various mathematical models. What is currently lacking, and especially in the field of hydroinformatics, is the study of such nonlinear dynamical systems (e.g. movement of the body of water in oceans, rivers, subsurface and surface, ecological processes, hydrometeorological processes etc.) through the methods and techniques developed in the theory of nonlinear dynamics. Often we know little about the structure of such complex dynamical systems, but in practice we can measure (partly) its output and some of its inputs. Consider, for example, the movement of a body of water in a particular location of the ocean near estuary: one can measure some limited output variables that are of practical interest, such as water levels, surges and currents. Furthermore, one could also measure (or estimate)

the forcing of this system, such as astronomical tides, wind speed and direction, air pressure, salinity, water temperature, discharges of a river etc. When we try to build a model of such a system, the ultimate goal is usually to establish the equations of motions, which will describe the underlying dynamics of the system in terms of meaningful quantities. Writing down the behaviour of the relevant components of the system in a mathematical language (e.g. Navier-Stokes equations in this case), we try to combine all we know about their actions and interactions. This approach allows us to construct a simplified model (image) of what happens in nature. Most of the knowledge we have about the inherent mechanisms has been previously derived from the first principles in mechanics (and of course under clear assumptions and limitations), though the relationships between various parameters involved in these equations, such as friction laws, diffusion mechanisms and turbulence mechanisms have been mostly derived from empirical observations. We usually call these models *physically-based* (or *process-based*). Alternatively, the theory of nonlinear dynamics and the concept of deterministic chaos allow for the construction of (learning, inducting) models that are based almost purely on time series data, produced by the dynamical system. These models learn the input-output relationships between the components from the observed data (time series), and are thus referred to as *data-driven* models.

The problem treated in this chapter lies at the heart of this work. What can we infer about the dynamical structure and laws governing the system under study, given a sequence of observations of one or a few time variable characteristics of the system, using nonlinear methods? We suppose that the domain knowledge about the dynamical system is limited to some assumptions that we may make about the general structure of these laws. Since the observations of the system are most likely incomplete, the solution—learning models from data—will not be unique (as we have discussed in Chapter 2). The ambiguity will be partially resolved by utilising the previous knowledge we posses about the system under study and by some physically-based restrictions that we can impose on the analysis. Thus the quest for a model consistent with the observations will be done employing the SRM induction principle, explained in previous Chapter 2, and the Bayesian approach (introduced in Chapter 2 and elaborated further in Chapter 4).

Models based entirely on time series data have the drawback that the terms they contain (such as relationships, weights, geometrical and dynamical invariants, transitional probabilities, parameters, etc.) do not usually have a meaningful interpretation, and thus are called sub-symbolic. *This lack of a physical interpretation is obviously not a failure of the individual methods employed, but is fundamental to this approach.* Nevertheless, a successful model learned from data (as a data and information, and eventually, knowledge encapsulator) can reproduce the data in a statistical sense, i.e. during simulation it yields a time series with a similar amount of information and properties as the original observed data. Furthermore, in particular modelling tasks, such as prediction, noise reduction, density estimation, and control, such models are often superior to the physically-based models. Ultimately, our goal is *to combine* the insight of the physically-based approach and intrinsic learning capabilities of the data-driven approach to modelling. This is to a large extent still an open problem. We advocate a sound mathematical framework together with the data-driven methods and techniques offered by the theory of nonlinear dynamics and deterministic chaos (evolving in the framework

of the SRM or Bayesian learning paradigm) as one possible skeleton for bridging the gaps between these two modelling approaches.

Bearing this in mind, in this chapter we describe the basics of the theory of nonlinear dynamical systems with special attention to the concept of deterministic chaos. As a support to the mathematical formulations of the nonlinear methods and techniques we use some solutions of well-known nonlinear dynamical systems and then further project the application of those techniques to real-life problems (demonstrated in Chapter 6). As we pointed out earlier in this text, even pure nonlinear deterministic systems can produce quite rich and irregular dynamic evolutions (solution of the system through time). Although we have not yet introduced the mathematical formulation and the methods of the theory of nonlinear dynamics and the concept of deterministic chaos we would like to demonstrate this observation by several examples.

*Example 3.1:*

Consider the analysis of the time series (output) found by numerical integration of the well-known Lorenz system in computational fluid dynamics. The goal of this experiment is to demonstrate the sensitive dependence of the solution of this system on the initial conditions. Lorenz (1963) studied a model of two-dimensional convection in a horizontal layer of fluid heated from bellow. He simplified the dynamics of the system to the following set of ordinary differential equations:

$$\frac{dx}{dt} = -\sigma x + \sigma y \qquad (3.1)$$

$$\frac{dy}{dt} = rx - y - zx$$

$$\frac{dz}{dt} = -bz + xy$$

where $x$ represents the velocity, $y$ is the fluid temperature difference, and $z$ is the deviation of the temperature from linear temperature profile at each instant, and $r$, $\sigma$, $b$ are positive parameters determined by the heating of the layer of fluid, the physical properties of the fluid and the height of the layer. These equation with three dependent variables and three parameters have a great diversity of solutions exhibiting complex dynamical structures (which cannot even be described in several pages here), and are most frequently utilised to study the origin and nature of deterministic chaos. The system (3.1) was numerically integrated using the fourth-order Runge-Kutta algorithm (see Parker and Chua, 1989) using the following values of the parameters $r$=27, $\sigma$=10, $b$=2.66, and with initial conditions $(x_0,y_0,z_0)$=(0,1,0). The time step used in the numerical integration was $\Delta t$=0.01 sec with 10000 iterations in total ($t$=100 sec). The second numerical integration was performed using slightly changed initial conditions $(x_0,y_0,z_0)$=(0,1.01,0) (differ by 0.01%). The solution of the variable $y(t)$ for both initial conditions is presented in Figure 3.1.

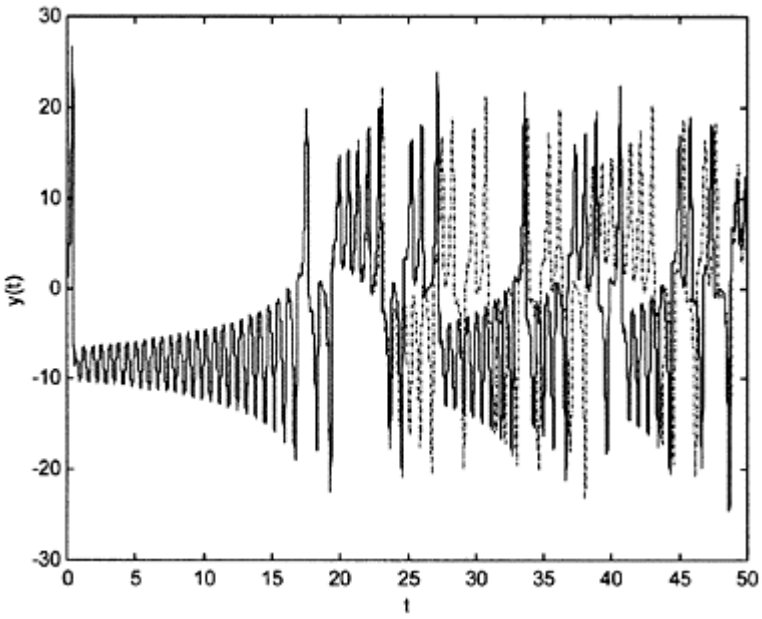



Figure 3.1. The graph of *y(t)* as a solution of Lorenz system (3.1) for $r$=27, σ=10, $b$=2.66, and $t$=0÷50 sec ($\Delta t$=0.01 sec). The solid line represents the solution for initial conditions (0,1,0) and the dashdotted line represents solution with initial conditions (0,1.01,0). The initial condition differ (0.01%).

As presented on Figure 3.1, the solution of the dynamical system (3.1) is highly sensitive on the choice of initial conditions. Due to the nonlinearity of the system, small perturbations and changes to the initial conditions can cause large differences in the response (output) of the system during its dynamical evolution. One could pose the following questions: Suppose the time series were measured as an outcome of a physical experiment (scale model) or real system. Can we, by analysis of the time series using nonlinear methods, infer the sensitivity to the initial conditions of this particular dynamical system?. What are the consequences? Can we quantify them?

*Example 3.2:*

Analysis of the time series generated (or measured) by numerical integration of the well

known Rössler (1976) system in computational mechanics. Our goal of this experiment is to demonstrate the existence of different irregular solutions of the system (different dynamical structures—regimes) with respect to change of some of the system's parameters.

$$\frac{dx}{dt} = -y - z \qquad (3.2)$$

$$\frac{dy}{dt} = x + ay$$

$$\frac{dz}{dt} = b + z(x - c)$$

As for the Lorenz system, the Rössler system (3.2) was integrated numerically using the fourth-order Runge-Kutta algorithm using the following values of the parameters: $a$=0.2, $b$=0.2, and the values of the parameter $c$ were increased from c=2.6, 3.5 to 4.1, using initial conditions ($x_0$, $y_0$, $z_0$)=(3, 0, 0). The time step used in the numerical integration was

$\Delta t$=0.01 sec with 10000 iterations in total ($t$=100 sec). The results, in the form of time series *x(t)*, the power spectra, and the phase portrait, of the solutions of the system using the three different values for the parameter $c$ are presented in Figure 3.2.

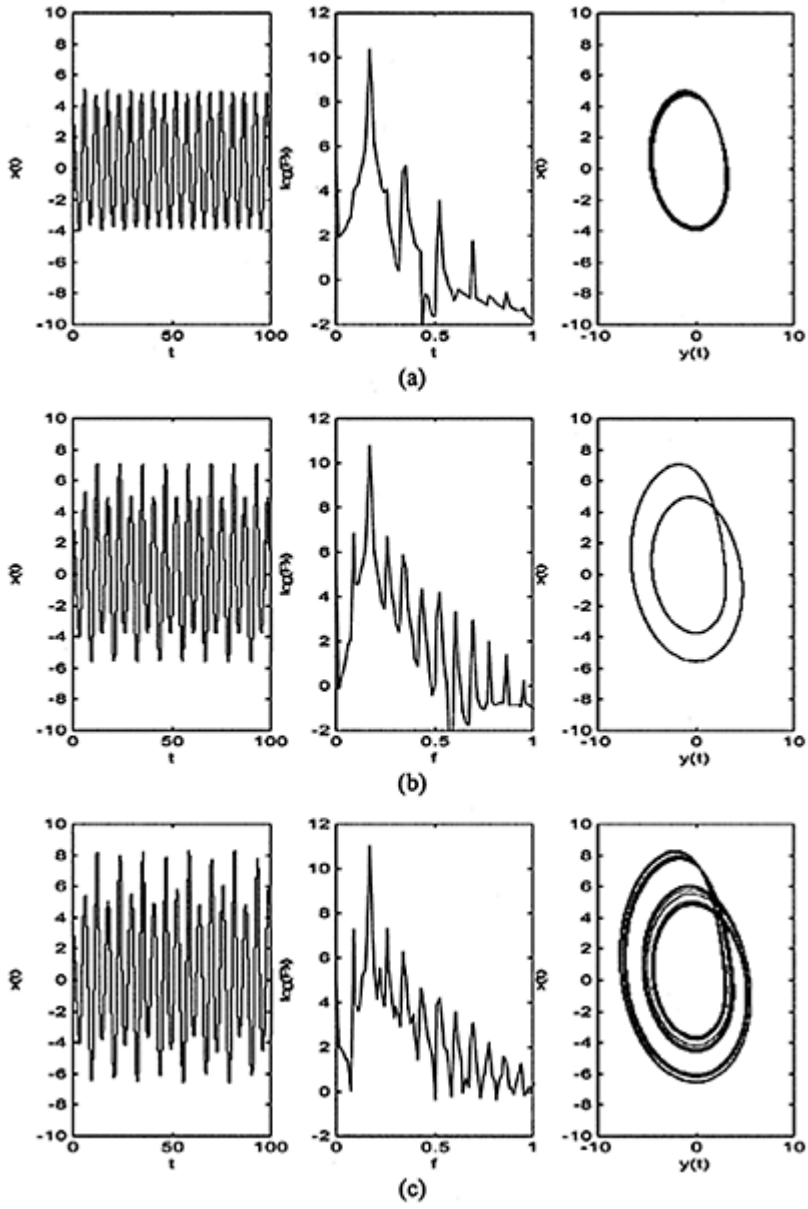



Figure 3.2. Solutions of the Rössler system (3.2) as time series *x(t)*, the power spectrum versus frequency and the phase portrait (projected plot of dependent variables *x(t)* and *y(t)*) for:

(a) $c$=2.6 a periodic; (b) $c$=3.5, a period-doubled; and (c) $c$=4.1, a period-quadrupled solution. Each orbit on the phase portraits moves clockwise as time $t$ increases.

The results presented in Figure 3.2 suggest that there is a periodic *attractor* (which will be discussed latter, but in this context it is a geometrical figure of the trajectory of the dynamical system) when $c$=2.6, that it has undergone a period doubling before $c$ increases to 3.5 and a quadrupling before $c$ reaches value of 4.1. One can see from the power spectra that the resultant noise (due to the round-off and truncation errors in both the numerical integration and analysis of the time series) dominates the signal at the basic level between the peaks. However, the peaks of the fundamental frequency and its harmonics are clearly identifiable in Figure 3.2(a). Also the appearance of subharmonics of order ½ is clearly visible in Figure 3.2(b), and the subharmonics of order ¼ in Figure 3.2(c). This is also confirmed by the phase portraits, which demonstrate the periodic, period-doubled and period-quadrupled (even the appearance of a broader band of frequencies) evolution of the dynamical system. A further increase of the value of the parameter $c$>4.5 leads to a highly irregular solution, which is characterised with appearance of a broadband spectrum, erratic signal and a strange geometrical figure in the phase portrait. As we will demonstrate latter, one speaks about onset of deterministic chaos. The results of this experiment arise the following questions: Suppose the time series were measured as an outcome of a real dynamical system. Can we, by analysis of the time series, identify existence of different dynamical regimes and presence of deterministic chaos? For which values of the control parameters and the nonlinear terms (components) the investigated dynamical system exhibits dynamical instabilities and routes to chaos? Can we model and predict such dynamical systems?

*Example 3.3:*

Numerical solution of the well-known Lotka-Volterra (or known as predator-prey) equations used in environmental modelling. The growth of a population of $x$ individuals of a species of prey and $y$ individuals of a species predator is governed by the equations:

$$\frac{dx}{dt} = x(a - cy) \qquad (3.3)$$

$$\frac{dy}{dt} = -y(b - cx)$$

where a, b, c >0 are reaction parameters. Lotka (1920) used these equations to model the chemical reactions in well-stirred conditions between different concentration of molecules $x$ and $y$. Voltera (1926) used predator-pray equations to model the population of fish in the Adriatic sea. These equation were numerically integrated using the fourth

order Runge-Kutta algorithm with the following values of the parameters *a*=?, *b*=?, *c*=? and using different initial populations. The resulting phase portrait together with the vector field is presented on Figure 3.3. The solution of this dynamical system exhibits reach geometrical and dynamical properties. Depending on the initial populations of the prey and predator, dynamical system may evolve asymptotically towards two possible equilibrium points (points where the vector field vanishes). According to the topological characters of the orbits near the equilibrium point, the solution of the system (see Figure 3.3) exhibits two types of equilibrium points (states), one is so-called *saddle point* (0,0) and the other is so-called *vortex* or *centre* (located in the first quadrant of the *(x,y)*-plane). The former one corresponds to a natural disaster, since both species will disappear from that particular aquatic environment, while the latter one is a favourable solution describing the desired balance between the both species. The mechanisms for generating such asymptotically

stable points will be elaborated latter in the context of the stability analysis of the dynamical systems. What is interesting for this dynamical system is the existence of so-called *basins of attraction*. There clearly exist two basins of attraction (geometrical spaces in the phase-space of the system), that is, given certain initial conditions and other geometrical and dynamical properties, the dynamical evolution of the system will take a path to one or other dynamical regime.

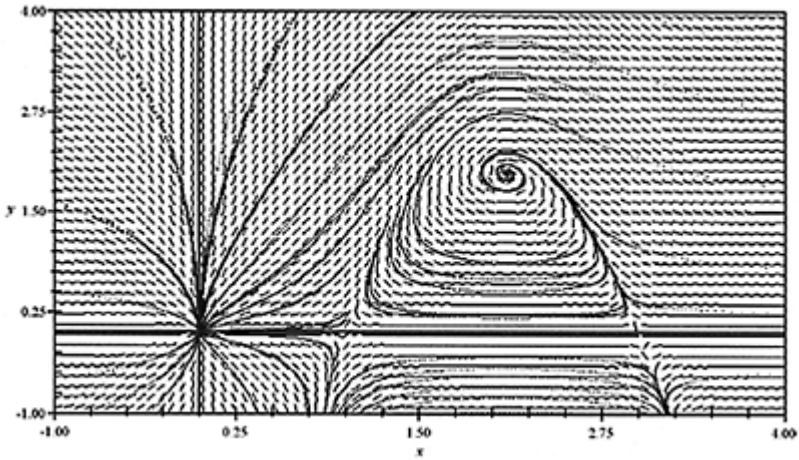



Figure 3.3. Phase portrait and vector field of the solutions of the predator-prey system (3.3) for different initial populations. Two asymptotic equilibrium points are clearly visible: (i) the saddle point (0,0) and (ii) the centre (vortex) point (2.125,2.125)

located in the first quadrant of the *(x,y)*—plane.

This example triggers the following questions: Given time series data observed on such a dynamical system, can we identify the existence of such equilibrium points and basins of attraction? How many are there? What are the conditions under which the dynamical system evolves towards one or another and even interchange between different attractors?

The main intention of these examples is to provoke some questions that we would like to explore and address throughout this work. Obviously, there is a broad range of questions that we have to address when talking about nonlinear dynamical systems and analysis of data produced by such systems. Furthermore, in a diverse field such as nonlinear dynamical systems, any selection of topics for a single chapter must be incomplete. However, the most important concepts for nonlinear analysis of time series and inducting models from those data are presented in this chapter. Analysis of data originating from nonlinear dynamical system is not as well established and is far from being well understood compared with its linear counterpart. In this chapter we make efforts to explain the perspectives and limitations of the data-driven methods based on the nonlinear dynamics we will introduce, which sometimes requires going back to the basics of physics and mathematics.

## 3.2 On dynamical systems

### *3.2.1 Essential definitions*

Any system whose time evolution from some initial state is described by a set of rules is called a *dynamical system*. When these rules are a set of differential equations, the system is called a *flow*, because their solution is continuous in time, whereas if the rules are set of discrete difference equations, the system is referred to as a *map*. The evolution of a dynamical system is best described in terms of its *phase space*, that is a coordinate system whose coordinates are all the variables that enter the mathematical formulation of the system. Thus the phase space can completely describe the state of the system at any moment in time. To each possible state of the system there corresponds a point in the phase space. The collection of all points in phase space that completely describes the dynamic evolution of the system is called a *trajectory*. The graph depicting the evolution of the system from different initial conditions is called a *phase portrait*. The final or asymptotically approaching equilibrium states are modelled by *limit sets*. By definition, a limit set that collects trajectories is called an *attractor* (we discuss latter the notion of an attractor in detail).

If the dynamical system is just a point particle of mass *m*, then its state at any given moment is completely described by its speed *v* and position *r*. Thus, its phase space is two dimensional with coordinates *v* and *r* or $p=mv$ and $q=r$, as in the common Newtonian notation. If instead we were dealing with a cloud of N particles, each of mass *m*, the phase space would be 2N-dimensional with coordinates $p_1, p_2,\ldots, p_n, q_1, q_2,\ldots, q_n$, where *N* indicates the number of independent positions or momenta or the number of degrees of

freedom. In classical mechanics the total energy $E$ of a dynamical system is related to the equation of motion via the *Hamiltonian H:*

$$H=H(q, p)=E_{kinetic}+E_{potential} \tag{3.4}$$

By definition, dynamical systems whose Hamiltonian does not vary in time are called *conservative systems*. Otherwise they are called *dissipative systems*.

The conservative systems have interesting properties, the most important of which is the *conservation of volumes in phase space*. For example, the conservation of mass of a fluid in motion is expressed analytically by the continuity equation:

$$\frac{\partial \rho}{\partial t} + \mathrm{div}\rho \boldsymbol{v} = 0 \tag{3.5}$$

where ρ is the density of the fluid at time $t$ and $\boldsymbol{v}$ is the velocity of the fluid at space point under consideration. Equation (3.5) holds for a motion of a fluid made by $N$-phase space points, provided that $\rho=\rho(q,p,t)$ stands for the density in phase space and $\boldsymbol{v}$ for the velocity of points in phase space (.i.e., the $2N$-dimensional vector with components $q(t)$ and $p(t)$). The term divρv can be written as:

$$\mathrm{div}\rho\boldsymbol{v} = \sum_{i=1}^{2N}\left(\frac{\partial \rho \dot{q}_i}{\partial q_i} + \frac{\partial \rho \dot{p}_i}{\partial p_i}\right) = \sum_{i=1}^{2N}\left(\dot{q}_i \frac{\partial \rho}{\partial q_i} + \dot{p}_i \frac{\partial \rho}{\partial p_i}\right) + \rho\sum_{i=1}^{2N}\left(\frac{\partial \dot{q}_i}{\partial q_i} + \frac{\partial \dot{p}_i}{\partial p_i}\right). \tag{3.6}$$

The second term on the right-hand side of (3.6) is equal to ρdivv. If we now recall from mechanics that for conservative systems $\partial H/\partial p_i = \dot{q}_i$ and $\partial H/\partial q_i = -\dot{p}_i$, one can find that div$\boldsymbol{v}$=0. Therefore (3.6) reduces to:

$$\frac{\partial \rho}{\partial t} + \sum_{i=1}^{2N}\left(\dot{q}_i \frac{\partial \rho}{\partial q_i} + \dot{p}_i \frac{\partial \rho}{\partial p_i}\right) = 0 \ . \tag{3.7}$$

Equation (3.7), known as Liouville's theorem, states that, in a $2N$-dimensional coordinate system $p_1, p_2,\ldots, p_n, q_1, q_2,\ldots, q_n$, $d\rho/dt$=0. Thus, since the mass is $m$=ρ$V$ (where $V$ denotes volume), one finds that for conservative dynamical systems $dV/dt$=0; volumes in phase space are conserved. This implies that for conservative systems the trajectories are constant-energy trajectories. Since each initial condition in phase space defines a unique constant-energy trajectory, the energy surface is the complete phase space, which bounds the volume. One can say that conservative systems cannot "forget" their initial states (or perturbations). A typical example for conservative system is frictionless motion of fluid, which is of course an idealistic case.

On the other hand, real dynamical systems are characterised by existence of various internal forces and processes (e.g. friction, shear stress, absorption and diffusion), and permanent interaction with other systems, thus, causing transformation of energy (e.g. in hydrodynamic motion of fluid transformation of kinetic energy into heat) and exchange of energy between interacting systems. Therefore, one has to deal with dissipative dynamical systems and observations produced by those systems in nature. One of the

most interesting properties of dissipative dynamical systems is the property of *irreversibility*, a property of most natural processes. This implies that the system does not remember any disturbance forever, thus irreversible processes are one-way evolutions of the dynamical systems. In mathematical terms, a process is reversible if it is indistinguishable when time is reversed, which is not the case with most physical processes in nature, and hence with processes related to water resources and aquatic environments. Another important property of the dissipative systems is the dissipation of the volume in phase space. The geometrical figures formed by the trajectories of such dynamical systems do not necessary reveal integer topological or Euclidean dimensions, but usually *fractal* or non-integer dimensions. Since the attractor cannot be a clear geometrical figure (e.g. point, circle, torus or hyper-tori), the only alternative is that the attracting set in question is a fractal set existing in a finite area of the phase space of zero volume. We demonstrate these characteristics using practical examples in the following sections.

### *3.2.2 Stability analysis of flows and maps*

In applications of the theory of nonlinear dynamical systems one is usually interested in enduring rather than transient phenomena, and so in steady states. Thus steady solutions of the governing equations are of special importance in order to further understand the transients and different evolutions of dynamical systems. Of these steady state solutions only the *stable* ones correspond to the states which persist in practice, and are usually the only ones observable. As we mentioned and demonstrated earlier, even pure nonlinear deterministic systems can produce a variety of solutions as a result of dynamical instabilities, that is, a small cause may have a large effect, or small disturbances at a given moment may grow and become significant such that after some time the behaviour of the system depends substantially on the nature of the disturbance, however small the disturbance was. Lorenz described this in a metaphor in which the unstable atmosphere might be triggered by the flutter of the wings of butterfly in a distant place, and thereby a devastating tornado may arise; this is the so-called *butterfly effect* in dynamical systems. A *bifurcation* occurs where the solution of a nonlinear system changes its qualitative character as parameters or some terms (components in the set of equations) change.

In this section our goal is to focus on the stability analysis of dynamical systems and to derive the necessary conditions for a map of the flow to be characterised as conservative or dissipative. Since the mathematical apparatus for stability analysis is well defined on linear systems, we first consider those systems, and than project the same reasoning to nonlinear dynamical systems.

Let us first consider a system of two linear, first-order ordinary differential equations (ODEs):

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 \\ \dot{x}_1 &= a_{21}x_1 + a_{22}x_2 \end{aligned} \tag{3.8}$$

where $a_{ij}$ are constants and $\dot{x}$ denotes $dx/dt$. Using vector notation (3.8) can be rewritten as

$$\dot{\mathbf{x}} = A\mathbf{x} \tag{3.9}$$

where $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$, $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and $\dot{\mathbf{x}} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix}$.

The equilibrium of the system (3.9) can be found if we set $\dot{\mathbf{x}} = 0$ or $A\mathbf{x}=0$. Therefore, if $A$ is nonsingular, the only equilibrium state is $\mathbf{x}=0$ ($x_1=0$ and $x_2=0$ in this example). Assume that a solution of (3.9) is of the form

$$\mathbf{x}(t)=\mathbf{c}e^{\lambda t} \tag{3.10}$$

where λ is a scalar and **c** is nonzero vector. Using (3.8), equation (3.10) can be rewritten as

$$A\mathbf{c}=\lambda\mathbf{c}. \tag{3.11}$$

A nontrivial solution to (3.11) for a given λ is the eigenvector, and λ is the eigenvalue. Since we are interested in a nontrivial (nonzero) solution, it is necessary that

$$Det(A-\lambda I)=0 \tag{3.12}$$

which usually called the determinant equation. This equation can be further written as:

$$Det\begin{pmatrix} a_{11}-\lambda & a_{12} \\ a_{21} & a_{22}-\lambda \end{pmatrix} = 0$$

$$(a_{11}-\lambda)(a_{22}-\lambda) - a_{12}a_{21} = 0$$

$$\lambda^2 + a_{11}a_{22} - a_{12}a_{21} - \lambda(a_{11}+a_{22}) = 0$$

or finally:

$$\lambda^2+\lambda TraceA+DetA=0. \tag{3.13}$$

This quadratic equation is the well-known characteristic equation. Its solutions $\lambda_1$ and $\lambda_2$ are eigenvalues which are either both real or both complex (complex conjugate). Once the eigenvalues are determined, the solution of (3.11) gives the corresponding eigenvectors:

$$\mathbf{c}_1 = \begin{pmatrix} c_{11} \\ c_{12} \end{pmatrix} \quad \text{and} \quad \mathbf{c}_2 = \begin{pmatrix} c_{21} \\ c_{22} \end{pmatrix}.$$

Assuming that $\lambda_1 \neq \lambda_2$, it follows that $\mathbf{c}_1$ and $\mathbf{c}_2$ are two linearly independent vectors in $\boldsymbol{R}^2$. Thus, both $\mathbf{c}_1 e^{\lambda t}$ and $\mathbf{c}_2 e^{\lambda t}$ are solutions to (3.10). Since the original system is linear, any linear combination of these solutions will also be a solution, or in general form:

$$\mathbf{x}(t) = a_1 e^{\lambda_1 t}\mathbf{c}_1 + a_2 e^{\lambda_2 t}\mathbf{c}_2 \ . \tag{3.14}$$

Let us now analyse the stability of such solution. Suppose that $\lambda_1$, $\lambda_2$ are real. If they are *both negative*, then in this case $\mathbf{x}(t) \to 0$ as $t \to \infty$ independently of the initial conditions $\mathbf{x}(0)$. Thus, in this case the evolution of the system is attracted to the equilibrium state no matter where the evolution initially started. One can say that the equilibrium state is *asymptotically stable*. This stable equilibrium state is called a fixed point or a node or an elliptic point. If $\lambda_1$, $\lambda_2$ are real *positive*, then one can infer that with $t \to \infty$, solution $\mathbf{x}(t) \to \infty$. In this case, regardless the initial conditions, the system *will not* approach the equilibrium state. Furthermore, even if the evolution of the dynamical system starts very close to the equilibrium state, it goes to infinity. In this case one can say that the origin repels all initial states, and it is thus *unstable*. If $\lambda_1 < 0 < \lambda_2$, we find that the contribution of $\lambda_1$ pushes the system towards the equilibrium state, whereas the contribution of $\lambda_2$ tries to repel further states from the equilibrium state. In other words, the linear combination of these two motions leads to evolutions that appear to approach the equilibrium state and then move away. One can define this unstable equilibrium state as a *saddle* (or a hyperbolic point).

If $\lambda_1$, $\lambda_2$ are complex with $\lambda_i = \alpha + \beta_i$, then the solution can be expressed as: $\mathbf{x}(t) = e^{\alpha t}(\mathbf{k}_1 \cos\beta t + \mathbf{k}_2 \sin\beta t)$, where $\mathbf{k}_1$ and $\mathbf{k}_2$ are appropriate vectors. If $\alpha$ is negative, then $x(t) \to 0$ as $t \to \infty$, and the equilibrium state is again asymptotically stable. If a is positive, then $\mathbf{x}(t) \to \infty$ as $t \to \infty$, and the equilibrium state is unstable. If $\alpha = 0$, then the solution is periodic where periodicity is determined by the initial conditions. This case is classified as *neutral stability* where the equilibrium state is called a *center* or a *vortex*.

From this analysis one can summarise that a dynamical system of $n$ first-order ordinary differential equations is asymptotically stable if the real parts of its eigenvalues are negative and it is unstable otherwise. If one recalls the characteristic equation (3.13) and its solutions (eigenvalues):

$$\lambda_{1,2} = \frac{TraceA \pm \sqrt{(TraceA)^2 - 4DetA}}{2} \tag{3.15}$$

one can easily prove that

$$\lambda_1 \lambda_2 = DetA$$
$$\lambda_1 + \lambda_2 = TraceA \tag{3.16}$$

If we combine (3.16) with our stability conclusions for this particular case ($n$=2), one can further conclude that the equilibrium state of the system is asymptotically stable if and only if $DetA > 0$ and $TraceA < 0$. In any other case the equilibrium state is unstable. For example, consider the system

$$\begin{aligned} \dot{x}_1 &= x_1 + 2x_2 \\ \dot{x}_1 &= 3x_1 + 2x_2 \end{aligned} \tag{3.17}$$

One can easily find the eigenvalues of the system: $\lambda_1=-1$ and $\lambda_2=4$. The $DetA=\lambda_1\lambda_2=-4$ and $TraceA=\lambda_1+\lambda_2=3$, thus the equilibrium state of dynamical system (3.17) is unstable.

We shall now extend the stability analysis to a *nonlinear system* of *n* first-order ODEs. In this case the system can be mathematically expressed as:

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2, ..., x_n)\\ \dot{x}_2 &= f_2(x_1, x_2, ..., x_n)\\ &\vdots\\ \dot{x}_n &= f_n(x_1, x_2, ..., x_n)\end{aligned} \tag{3.18}$$

where $f_1$, $f_2$ and $f_n$ are nonlinear functions of all or some of the variables $x_1,\ldots, x_n$. When these functions do not depend explicitly on time, hence there are no time-varying external forces acting on the dynamical system and the flow field **f** is stationary, such dynamical systems are called *autonomous*. Since the right-hand side of (3.18) is stationary, it can be proven that no two trajectories (corresponding to two evolutions from two different conditions) will intersect in the phase space (see for example Rosen, 1970). If on the other hand, the functions in (3.18) $f_1$, $f_2$ and $f_n$ depend explicitly on time, the dynamical system is called *nonautonomous*.

As analytical solution of (3.18) is usually not obtainable and a straightforward application of the stability analysis described above is not applicable. To address the issue of stability in this case, we have to proceed with an analysis which investigates the properties of system for $x_1 = \bar{x}_1 + x_1^{'}, \quad x_2 = \bar{x}_2 + x_2^{'}, \ldots, \quad x_n = \bar{x}_n + x_n^{'}$, where $x_i^{'}$ indicate very small deviations from an equilibrium state $(\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$. The system (3.18) can be rewritten as:

$$\begin{aligned}\frac{d(\bar{x}_1 + x_1^{'})}{dt} &= f_1(\bar{x}_1 + x_1^{'}, \bar{x}_2 + x_2^{'}, ..., \bar{x}_n + x_n^{'})\\ \frac{d(\bar{x}_2 + x_2^{'})}{dt} &= f_2(\bar{x}_1 + x_1^{'}, \bar{x}_2 + x_2^{'}, ..., \bar{x}_n + x_n^{'})\\ &\vdots\\ \frac{d(\bar{x}_n + x_n^{'})}{dt} &= f_n(\bar{x}_1 + x_1^{'}, \bar{x}_2 + x_2^{'}, ..., \bar{x}_n + x_n^{'})\end{aligned} \tag{3.19}$$

One can simplify the above system of differential equations by ignoring all nonlinear terms involving fluctuations on the right-hand side (equal zero). In this way one can effectively replace $f_1(\bar{x}_1 + x_1^{'}, \bar{x}_2 + x_2^{'}, ..., \bar{x}_n + x_n^{'})$ by $f_1^{'}(x_1^{'}, x_2^{'}, ..., x_n^{'}) = a_{11}x_1^{'} + a_{12}x_2^{'} + ... + a_{1n}x_n^{'}$, etc.

Considering that $d\bar{x}/dt = 0$, the system (3.19) becomes

$$\begin{aligned}
\dot{x}_1^{'} &= a_{11}x_1^{'} + a_{12}x_2^{'} + \ldots + a_{1n}x_n^{'} \\
\dot{x}_2^{'} &= a_{21}x_1^{'} + a_{22}x_2^{'} + \ldots + a_{2n}x_n^{'} \\
&\vdots \\
\dot{x}_n^{'} &= a_{n1}x_1^{'} + a_{n2}x_2^{'} + \ldots + a_{nn}x_n^{'}
\end{aligned} \tag{3.20}$$

which is in fact a linear system of first-order differential equations describing the evolution of the *fluctuations* $(x_1^{'}, x_2^{'}, \ldots, x_n^{'})$ about the equilibrium state $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$. If the fluctuations grow in time then the system is driven away from the equilibrium state and is unstable, whereas otherwise is stable. In a vector form, the system (3.2) can be presented as:

$$\dot{\mathbf{x}}^{'} = A\mathbf{x}^{'} \tag{3.21}$$

where

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

The stability of the equation (3.21) is, as described before, determined by the eigenvalues $\lambda_i$, defined by the equation

$Det(A-\lambda I)=0$

where the matrix A in this case is the Jacobian matrix of **f** evaluated at $\bar{\mathbf{x}}$. This is a direct result of the application of the Taylor's theorem, stating that a nonlinear function $f(x_1, x_2, \ldots, x_n)$ is equal to

$$f(x_1, x_2, \ldots, x_n) = f(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n) + x_1^{'}\left.\frac{\partial f}{\partial x_1}\right|_{x_1=\bar{x}_1,\ldots,x_n=\bar{x}_n} + x_2^{'}\left.\frac{\partial f}{\partial x_2}\right|_{x_1=\bar{x}_1,\ldots,x_n=\bar{x}_n} + \ldots + x_n^{'}\left.\frac{\partial f}{\partial x_n}\right|_{x_1=\bar{x}_1,\ldots,x_n=\bar{x}_n} + h.o.t.$$

Knowing that at equilibrium $f(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n) = 0$ and by neglecting the higher-order terms one obtains:

$$f(x_1, x_2, \ldots, x_n) = x_1^{'}\left.\frac{\partial f}{\partial x_1}\right|_{x_1=\bar{x}_1,\ldots,x_n=\bar{x}_n} + x_2^{'}\left.\frac{\partial f}{\partial x_2}\right|_{x_1=\bar{x}_1,\ldots,x_n=\bar{x}_n} + \ldots + x_n^{'}\left.\frac{\partial f}{\partial x_n}\right|_{x_1=\bar{x}_1,\ldots,x_n=\bar{x}_n}$$

Since $\dot{x} = f(x_1, x_2, ..., x_n)$, for a nonlinear system of *n* first-order ODEs we have:

$$\dot{\mathbf{x}} = A'\mathbf{x}'$$

and further taking into account that $\dot{x} = \dot{\bar{x}} + \dot{x}'$ and $\dot{\bar{x}} = 0$, finally we have:

$$\dot{\mathbf{x}} = A'\mathbf{x}' \tag{3.22}$$

where

$$A' = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \frac{\partial f_3}{\partial x_n} \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}. \tag{3.23}$$

If one compares the equations (3.22) and (3.21), than it follows that $A=A'$ This result provides us in many cases the possibility to investigate the stability of equilibrium states of dynamical systems. Important is to note that it is always the eigenvalues computed from $Det(A-\lambda I)=0$ or $Det(A'-\lambda I)=0$ that determine the stability of a dynamical system. We discuss latter the physical interpretation of the values of these eigenvalues.

The dynamical systems we have considered up to this point are described by a set of differential equations that have continuous solution and we referr to them as flows. To emphasise that there are nonlinear systems other than flows, we shall next introduce *nonlinear difference equations*. Difference equations are mathematically interesting and have many important applications. They are also, in many ways, more elementary and from application point of view more fundamental than differential equations due to several reasons. Firstly, they involve a discrete variables (ones that can be mostly observed on real physical systems) rather then continuous ones. Secondly, the solution of complex nonlinear dynamical systems described by sets of differential equations (such as the systems we model in hydraulics and hydrology) can be found almost exclusively by numerical integration, and thus, one has to deal with sets of difference equations.

A *difference equation*, *recurrence equation* or *map* is in general of the form:

$\mathbf{x}_{n+1}=\mathbf{F}(\mathbf{x}_n, n)$ for $n=0,1,2,\ldots, N$ (3.24)

where $\mathbf{x}_n \in \boldsymbol{R}^m$ and functions $\mathbf{F}: \boldsymbol{R}^m \times \boldsymbol{Z} \rightarrow \boldsymbol{R}^m$, which depend on certain parameters and time. Thus difference equations are functional, sometimes algebraic, systems that

correspond to differential equations. The integral variable *n* here corresponds to the independent real variable t. For example, the dynamical system

$$\begin{aligned} x_{n+1} &= 1 - ax_n^2 + y_n \\ y_{n+1} &= bx_n \end{aligned} \tag{3.25}$$

is a two-dimensional map known as Hénon map (Hénon, 1976), while the dynamical system

$$x_{n+1}=\mu x_n(1-x_n) \tag{3.26}$$

is one-dimensional system known as logistic equation (May, 1976) frequently used to model the growth of a certain population in ecosystems.

Stability analysis on maps is very similar to stability analysis on flows. One takes $\mathbf{x}_0$ as the given initial conditions of the dependent variables and considers their dynamical evolution, as $n\to\infty$. Furthermore one can study the functions **F**, which depend on certain parameters, and how the solutions $\{\mathbf{x}_n\}$, i.e. the sequences $\{\mathbf{x}_0, \mathbf{x}_1,\ldots\}$, change both quantitatively and qualitatively with those parameters. Consider the stability of the Hénon map introduced above (3.25). The equilibrium points (or fixed points) are found by assuming that $x_{n+1} = x_n = \bar{x}$ and $y_{n+1} = y_n = \bar{y}$ (where similarly $\dot{x} = 0$ ,$x_{n+1}-x_n=0$).

From (3.25) at equilibrium point one can write

$$\begin{aligned} \bar{x} &= 1 - a\bar{x}^2 + \bar{y} \\ \bar{y} &= b\bar{x} \end{aligned}$$

which results in second-order algebraic equation $a\bar{x}^2 + (1-b)\bar{x} - 1 = 0$ ,and thus the two fixed points are:

$$\left( \bar{x}_1 = \frac{(b-1) + \sqrt{(b-1)^2 + 4a}}{2a}, \bar{y}_1 = b\bar{x}_1 \right)$$

$$\left( \bar{x}_2 = \frac{(b-1) - \sqrt{(b-1)^2 + 4a}}{2a}, \bar{y}_2 = b\bar{x}_2 \right)$$

As with flows, one can proceed by investigating the properties of a map in the presence of small fluctuations about the equilibrium state, that is, $x = \bar{x} + x'$, $y = \bar{y} + y'$. Equation (3.25) then becomes

$$\begin{aligned} \bar{x} + x'_{n+1} &= 1 - a(\bar{x} + x'_n)^2 + \bar{y} + y'_n \\ \bar{y} + y'_{n+1} &= b\bar{x} + bx'_n \end{aligned} \tag{3.27}$$

Neglecting the terms involving fluctuations higher than first order, (3.27) can be written as

$$x'_{n+1} = -2a\bar{x}x'_n + y'_n$$
$$y'_{n+1} = bx'_n$$

which now represents a set of two linear first order difference equations. It can be written in vector form as

$$\begin{pmatrix} x'_{n+1} \\ y'_{n+1} \end{pmatrix} = A \begin{pmatrix} x'_n \\ y'_n \end{pmatrix} . \qquad (3.28)$$

where $A = \begin{pmatrix} -2a\bar{x} & 1 \\ b & 0 \end{pmatrix}$. If one applies the previously derived conditions for stability of flow (for set of 2 equations), the Henon map is stable if $DetA=-b>0$ and $TraceA = -2a\bar{x} < 0$, and unstable otherwise. It is also obvious that $A$ represents the Jacobian of **F** at $\bar{\mathbf{x}}$ for $f_1(x, y)=1-2ax^2+y$ and $f_2(x, y)=bx$.

The equation (3.28) can be extended to system of $n$ difference equations and represents a very interesting property for deriving the necessary conditions for a map to be characterised as *conservative* or *dissipative* system. Thus, *for maps,* by regarding a set of perturbations as defining some initial volume in phase space, one can conclude that this volume will not grow or decay (i.e. it will be conserved) if $|DetA|=1$. On the other hand, *for flows,* whether their volumes in phase space defined by the trajectories expand or contract or remain same is determined by the *trace of A* and not by the determinant. A flow evolution of a perturbation is given by (3.21). This equation has a solution $\mathbf{x}'(t) = e^{tA}\mathbf{x}'(0)$, where $\mathbf{x}'(0)$ is the initial vector. Assuming that $A$ has distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ one can find matrix $U$ such that $U^{-1}AU=D$, where $D$ is diagonal:

$$D = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

$A$ can be written as

$A=UDU^{-1}$.

Using the multiplication theorem, we derive that

$DetA=(DetU)(DetD)Det(U^{-1})=DetD=\lambda_1\lambda_2\ldots\lambda_n$

(3.29)

Similarly,

$$TraceA=TraceD=\lambda_1+\lambda_2,+\ldots+\lambda_n \tag{3.30}$$

In fact, one could generalise the above to consider not just but any function *f(A):*

$$Det[f(A)]=Det[f(D)]=f(\lambda_1)f(\lambda_2)\ldots f(\lambda_n) \tag{3.31}$$

and

$$Trace[f(A)]=Trace[f(D)]=f(\lambda_1)+f(\lambda_2),+\ldots+f(\lambda_n). \tag{3.32}$$

Assuming that $f(A)=e^{At}$, the determinant (3.31) becomes:

$$Det(e^{At}) = e^{\lambda_1 t}e^{\lambda_2 t}\cdots e^{\lambda_n t} = e^{(\lambda_1+\lambda_2+\ldots+\lambda_n)t} = e^{(TraceD)t} = e^{(TraceA)t} \tag{3.33}$$

Now, we can similarly argue that a volume $V$ of perturbations in phase space will be conserved if $|Det(e^{At})|=1$, which translates to $|e^{(TraceA)t}|=1$ or $TraceA=0$. Thus, a flow represents a conservative system if the trace of the Jacobian is zero and a dissipative system if $|e^{(TraceA)t}|<1$, which translates to $Trace\ A<0$. Since $TraceA=\lambda_1+\lambda_2+\ldots+\lambda_n$, the sum of the eigenvalues dictates whether volumes contract or expand or remain the same. It is important to stress that each eigenvalue gives the rate of contraction or expansion along a direction of one of the coordinates in phase space. If all eigenvalues are negative the volumes contract along all directions. Obviously, one can have positive and negative $\lambda$'s, while $\lambda_1+\lambda_2+\ldots+\lambda_n<0$. It is thus possible to have expansion along certain directions in phase space, even though the initial volume shrinks in time. Such systems will be latter termed as *deterministic chaotic systems*. The eigenvalues $\lambda_1, \lambda_2,\ldots, \lambda_n$, are known as *Lyapunov exponents* of the flow (Eckman and Ruelle, 1985). Direct extension of the above arguments to maps where $\mathbf{x}_{n+1} = A\mathbf{x}_n$, leads to the conclusion that the Lyapunov exponents are the logarithms of the eigenvalues of *A*. Note that in dissipative systems even though an initial volume shrinks to zero it does not necessarily mean that the volume will shrink to a point. In a 3D phase space a surface (e.g. the geometrical figure of the attractor) has zero volume, but it is not a point.

Finally, let us illustrate the above arguments for the stability of nonlinear dynamical systems on the examples already introduced earlier. For the Lorenz system (3.1) where $r,\sigma,b$ are positive physical parameters, one can conclude that it is a dissipative system, since

$$A=\begin{pmatrix} -\sigma & \sigma & 0 \\ -z+r & -1 & -x \\ y & x & -b \end{pmatrix}$$

and $TraceA=-(\sigma+1+b)<0$. For the Rössler system (3.2) where $a,b,c$ are positive constants, the trace of the Jacobian $A$ is $TraceA=a-c-x$ and the system is dissipative

when $a<c$. For the Hénon map (3.25), one can find that it corresponds to dissipative system if $|DetA|=|b|<1$.

### *3.2.3 Attractors and strange attractors*

The solution of a nonlinear dynamical system from some initial conditions with the presence of small perturbations can in general evolve mathematically following three possibilities: (i) the system can be "attracted" by the stable equilibrium state; (ii) repelled by the unstable equilibrium state; and (iii) engaged in a never-ending motion (frictionless systems). For all these mathematical possibilities in the real world only the first one is plausible. The settling part or the *transient* of the system is modelled by the trajectory in phase space. The final state or the equilibrium state is modelled by limit sets. In the previous section we have already introduces four such limit sets, and all of them points: fixed point, repeller, center and saddle point. Furthermore we defined the attractor as a limit set which collects trajectories, that is, the different trajectories describing the system transients from different geometrical objects (e.g. point limit set). Consider, for example, the standard seiche test performed for any numerical model in computational hydraulics. Without bottom or wall friction (free-slip boundary conditions) the motion of the fluid is simply a conservative system whose evolution is undisturbed. When bottom (and/or wall friction) is introduced, the kinetic energy is gradually being spent and eventually the motion of the fluid stops at the stable equilibrium point (in phase space). Points, however, are not the only limit sets. A cycle or an ellipse may also be a limit set for a trajectory of dynamical system. An example of such an attractor, which describes regular periodic motion, was already given in the Example 3.2 and is presented on Figure 3.2a. When the system is disturbed, its intrinsic dynamics soon assembles regular periodic motion with a particular frequency. Figure 3.4 schematically presents this type of attractor.

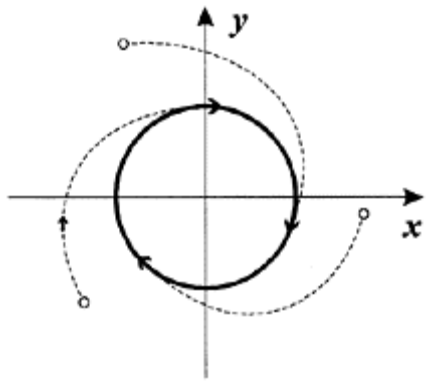



Figure 3.4. Phase portrait of a two-dimensional system having a limit cycle as an attractor. Trajectories from different initial conditions are attracted

and stay on the cycle. The evolution of the system is periodic.

A periodic attractor of a limit cycle may be embedded in more than two dimensions. In such cases the trajectory may form complex geometrical shapes, which, when projected might give an impression that the trajectory intersects itself, which is not the case (see Figure 3.5). If there exist several such periodic components in the motion of dynamical system, the total evolution of the system is quasi-periodic. In such motion the trajectory fills the surface of a torus in three-dimensional phase space and a hyper-torus in phase space defined with more than three dimensions. In this sense, quasi-periodic motions can look quite "irregular".

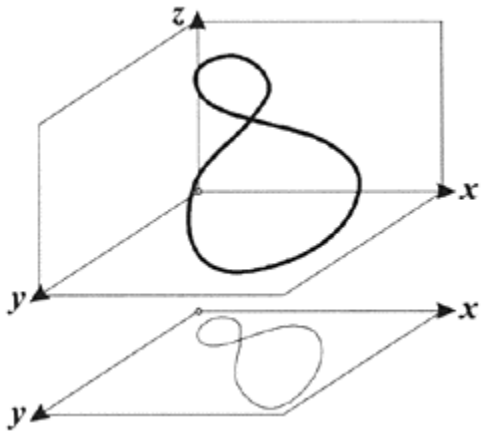



Figure 3.5. A periodic attractor in 3D phase space and its plane phase portrait.

Quasi-periodic motions occur in nature quite often. For example, astronomical tidal motion is composed of more than hundred such periodic components, due to the relative motions of earth, sun and moon. Ideally, daily temperature can be viewed as a quasi-periodic variable with several distinct frequencies. The previous Figure 3.2.c shows a projected phase portrait of such quasi-periodic motion.

Up to this point we have discussed three types of attractors: points, limit cycles and tori. All of them are submanifolds of the total available phase space. In addition, they are topological structures characterised by topological integer dimensions of 0,1,2,…, *n*, or by Euclidean dimensions of 1,2,3,…, *n*+1, respectively. The identification of these attractors from dynamical systems is quite straightforward. Linear methods, such as, Fourier analysis can verify if a given evolution is steady state, periodic, or quasi-periodic (see Example 3.2). A very interesting modelling property of nonlinear dynamical systems that exhibit such attractors is that *the long-term predictability of these systems is guaranteed*. When a dynamical system exhibits a torus as its attractor, a set of different

initial conditions defines a set of different trajectories each of them assembling around the torus, gradually filling its surface but without diverging from one another. This provides some kind of long-term predictability, since small perturbations will not grow; see Figure 3.6. Due to the small irregularities and perturbations, the future values may differ from the observations. However, *data assimilation techniques* can be employed in this case to improve the predictions.

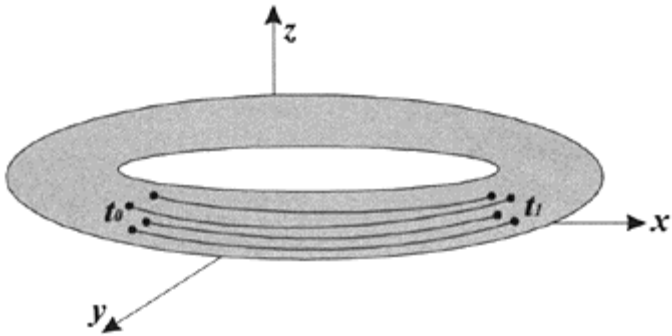



Figure 3.6. A torus as an attractor of a dynamical system. Trajectories defining the evolution of the system from different initial conditions do not diverge. A long-term predictability is ensured.

Let us now consider the power spectrum of some output of a nonlinear dynamical system presented on Figure 3.7. The power spectrum is distributed over a wide range of frequencies that have almost the same contribution, thus generating a *broadband* power spectrum. Such spectra are indicative of nonperiodic random evolutions where there is motion in all frequencies (even not really distinguishable from a white noise).

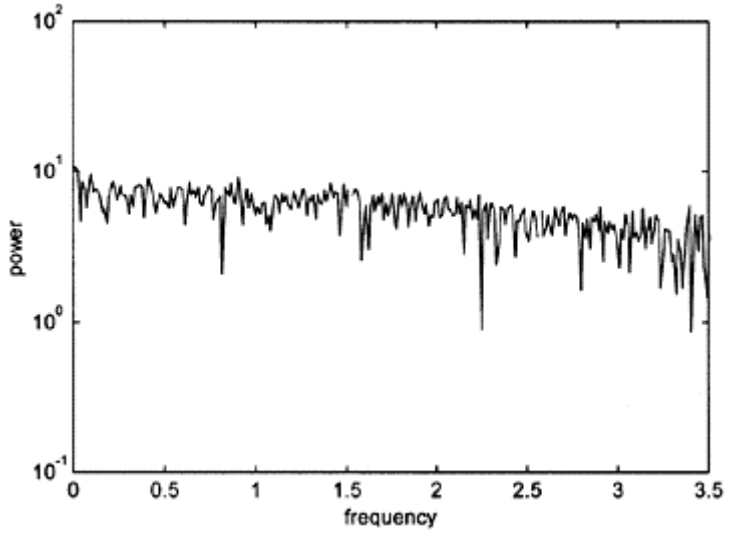



Figure 3.7. Power spectrum showing contribution of all frequencies. Such spectra are often called broadband noise spectra and are indicative of random motion.

The question normally follows: Is it possible that such spectra can be generated by a deterministic dynamical system? Let us assume that the answer is yes. This implies that the evolution of such dynamical system must be described by a nonperiodic trajectory in a phase space which never intersects itself. Thus, the trajectory must be of infinite length and confined in a finite area of the phase space of zero volume. Since the attractor cannot be a torus or hyper-torus (then the trajectory is of infinite length and confined to a finite area), the only possible alternative is that the attracting set in question is fractal set (Parker and Chua, 1989), which exhibits a non-integer dimension. The first such dynamical system was discovered by Lorenz (1963) from the convection equations (see Example 3.1). The power spectrum presented on Figure 3.7 is estimated on the variable *y(t)* as a solution of Lorenz system (3.1). The attractor of the Lorenz system in different views is presented on Figure 3.8, revealing an “interesting” nonplanar geometrical shape and nonintersecting nearby orbits of the trajectory.

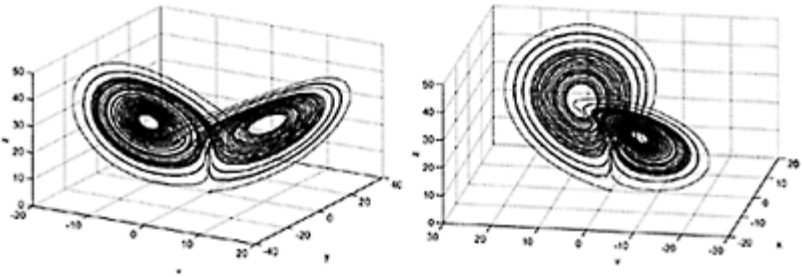


Figure 3.8. Different views of the trajectory and the attractor of the Lorenz system (3.1). The evolution of the system is described by a trajectory which is 10000 time steps long (100 seconds, since $\Delta t$=0.01 sec).

It is obvious that the Lorenz attractor does not look like the topologically well-shaped attractors previously described. The trajectory is deterministic since it is the result of the solution of the nonlinear dynamical system, previously described in Example 3.1. However this trajectory is strictly nonperiodic. The simulation of the trajectory shows that it loops and jumps from one part of the attractor to the other irregularly. Intensive studies of the Lorenz attractor have shown that the fractal dimension of the attractor is estimated to be about 2.06 (see, for example, Grassberger and Procaccia, 1983). However, the fractal nature of the attractor does not merely mean nonperiodic orbits; It also causes nearby trajectories to diverge. The trajectories which are initiated form different initial conditions soon reach the attracting set, but two nearby trajectories do not stay close to each other after some time. They soon diverge and follow totally different paths in the attractor. This divergence means that the evolution of the dynamical system from two slightly different initial conditions will be completely different, thus implying a sensitive dependence on initial conditions, as we demonstrated in Example 3.1 and Figure 3.1. If we “restart” the system from various initial conditions all of the resulting trajectories will be bound to the attracting set. However, qualitatively all of the solutions differ. In this case one can say that the system has generated randomness. We can now see that there exists a nonlinear dynamical system, even though it can be described by simple deterministic rules (differential equations) that can generate such deterministic randomness, which is usually termed *chaos*. These dynamical systems are called *deterministic chaotic systems*, and their attractors are called *strange attractors*. Some other examples of strange attractors of the dissipative dynamical systems that we have introduced earlier are shown on Figure 3.9 and Figure 3.10.

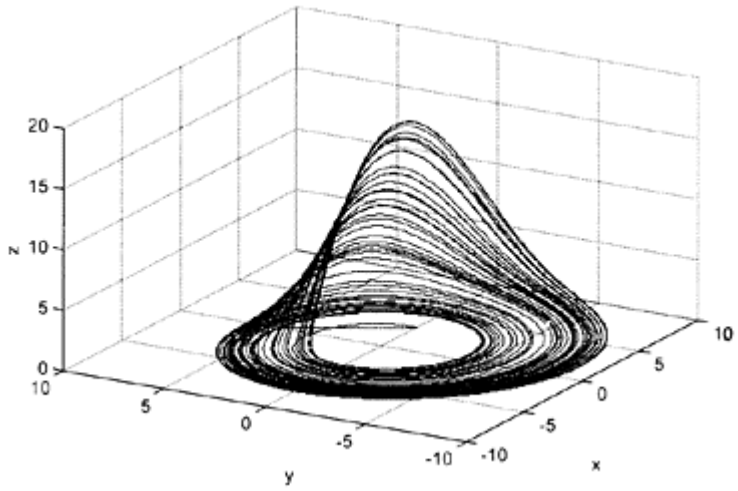



Figure 3.9. The strange attractor of the Rössler system (3.2). The evolution of the system is described by a trajectory which is 20000 time steps long ($\Delta t$=0.01 sec).

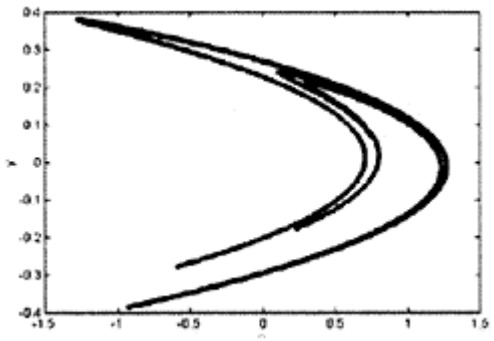



Figure 3.10. The strange attractor of the Henon map (3.25), $a$=1.4, $b$=0.3. The evolution of the system is described by a trajectory which is 15000 iterations long.

*3.2.4 Delineating and quantifying the dynamics*

In this section we shall briefly introduce some of the key concepts for delineating and quantifying the dynamics of nonlinear systems (still seen as a set of differential equations). A separate section will be devoted for reconstruction and modelling of the underlying dynamics from observations, since inducting models from data is the key focus of this work.

## POINCARÉ SECTIONS

A classical convenient technique for delineating the dynamics of a system is given by the Poincaré sections or maps. It replaces the flow of an *n*th-order continuous-time system with an (*n*-1)th-order discrete-time system. The resulting map is thus called a *Poncaré map*, ensuring the its limit sets correspond to the limit sets of the underlying flow. The usefullnss of the Poncaré map lies in the reduction of order of the dynamical system and the fact that it bridges the gap between continuous and discrete-time systems. The simplest way to describe the Poincaré map is that it represents a slice through the attractor of the dynamical system. First from a suitable oriented surface in *m*-dimensional phase space (see Figure 3.11) one can construct a map on this surface by capturing the trajectory of the flow. The iterates of the map are given by the points where the trajectory intersects the surface in a specified direction (from above in Figure 3.11). Thus the map checks every full orbit around the attractor.

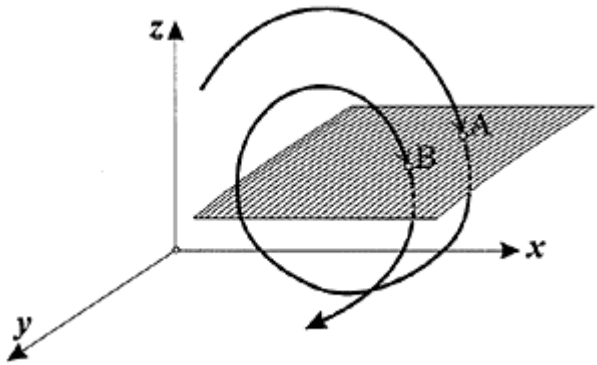



Figure 3.11. Poincaré section (map) of a flow in three dimensions. The successive intersection points A, B,…of the continuous trajectory with the surface of sections define iterates of a two-dimensional map in this case.

It is important to stress that the discrete "time" of the Poincaré map is the intersection count and is usually not simply proportional to the original time *t* of the flow. The time a trajectory spends between two successive intersection points will vary, depending both on the type of the trajectory (dynamics) and on the surface of the section chosen. If one deals

with periodic evolution of period *n*, then this sequence consists of *n* dots repeating in the same order (see Figure 3.12a). If the evolution of the system is quasi-periodic the sequence of points defines a closed limit cycle (see Figure 3.12b). Finally, if the evolution is deterministic chaos, then the Poincaré section is a collection of points that show an interesting pattern, often revealing the fractal nature of the underlying attractor (see Figure 3.13).

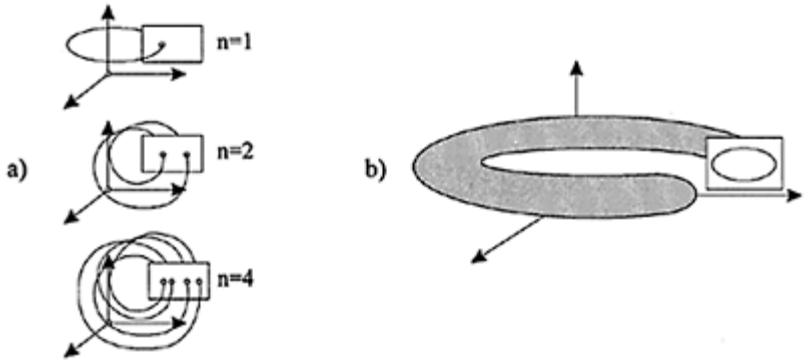



Figure 3.12. (a) Periodic evolution of dynamical system and the Poincaré section. (b) A quasi-periodic evolution of a dynamical system. The Poincaré section is a sequence of points defining a limit cycle.

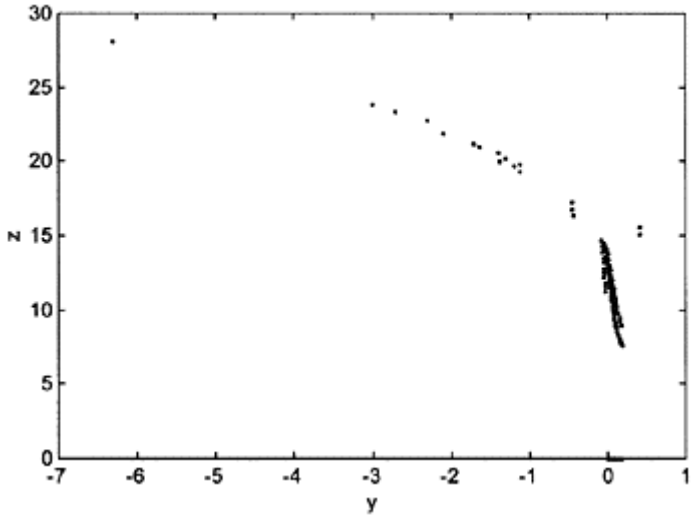


Figure 3.13. Poincaré section of the Lorenz attractor. The *(y, x)* variables are plotted every time the *x*-variable equals zero and its derivative is negative.

The process of obtaining the Poincaré section corresponds to sampling the phase space of the dynamical system in order to reveal some of its dynamics. Thus, the choice of the proper surface of section will then be a crucial step in the analysis of data. However, in many cases the surface of sections, and therefore the appropriate sampling interval, can be defined in such a way that it corresponds to a physically meaningful measure of the dynamical system. For example, for a periodically or quasi-periodically forced motion (e.g. tidal water levels and currents) one can sample the trajectory at times that are multiple integers of the forcing periods. Then the sequence of strictly compatible points is obtained, which can be further analysed. Another interesting application of the Poincaré section could be the collection of all minima or all maxima of the dynamical system if one is interested in modelling the dynamics of extreme events. Collecting maxima (or minima) corresponds to performing a section by a surface of the zero time derivative (as in case of Figure 3.13).

Furthermore, analysis and modelling of the sequence of times between successive intersections could expose some interesting properties of the underlying dynamics. More generally, the times between successive passages of a continuous trajectory through a Poincaré section are related deterministically to the properties of the motion in between. An individual time interval is given by the length of the path from one intersection to the next divided by the average velocity of the phase space vector on this path. Therefore, it is plausible that the sequence of time intervals obtained from the Poincaré section allows the reconstruction of the deterministic motion. These kinds of application of the Poincaré section are demonstrated latter on in case-study applications (see Chapter 6).

Finally, when dealing with dynamical systems whose attractors live in low-dimensional phase space (e.g. 3D, but will be discussed latter), performing Poincaré surface of sections can be used to establish approximate relationships between the variables, i.e. the coordinates defining that phase space. For example, Figure 3.14 shows the Poincaré section of the Rössler attractor defined with the plane $y=0$. One can see the relationship between the variables $x$ and $z$ when $y=0$.

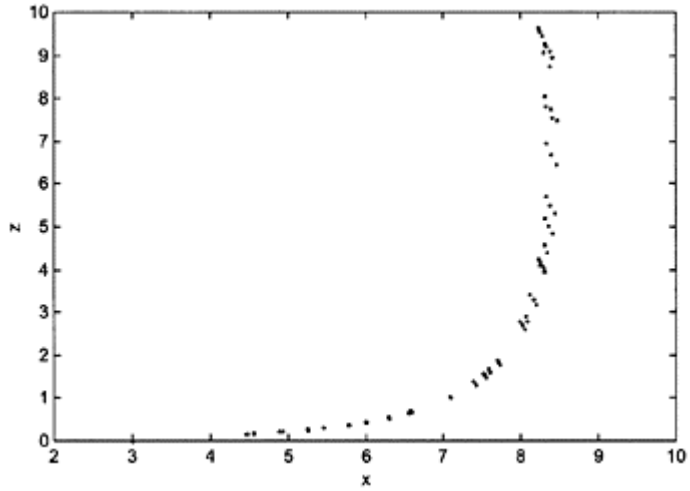
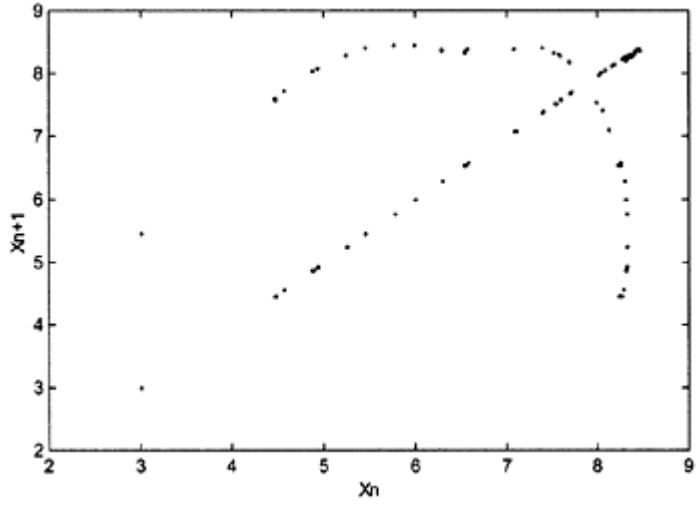



Figure 3.14. Poincaré section of the Rössler attractor with plane $y$=0 (see also Figure 3.9). One could establish simple relationship between the variables $x$ and $z$.

These kinds of examples can also trigger a further approximation of the two-dimensional section by a one-dimensional mapping, simply by recording a sequence of values of particular variable (e.g. $x$) for successive intersections, and then plotting $x_t$ versus $x_{t+1,}$ see Figure 3.15. In this way, one could obtain the *return map*.



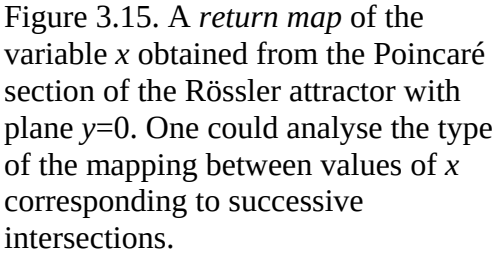


Figure 3.15. A *return map* of the variable $x$ obtained from the Poincaré section of the Rössler attractor with plane $y$=0. One could analyse the type of the mapping between values of $x$ corresponding to successive intersections.

## DIMENSIONS

In the previous sections we discussed the dynamical side of deterministic chaos which manifests itself in the sensitive dependence of the evolution of a dynamical system on its initial conditions and small perturbations. This strange behaviour in time of a deterministic chaotic system has its counterparts in the geometry of the limit set in phase space formed by the trajectory of the system, the attractor. Thus, one could say that the dynamics of a system are dictated by the geometry of the phase space and its attractor. This geometry can be quantified by a series of geometrical and dynamical invariants, termed as dimensions and Lyapunov exponents. In this section we show that the strange attractors possess non-integer dimensions while the dimension of a non-chaotic attractor is always an integer. Furthermore, the notion of dimensions allows for a remarkable property that an attractor can be identified and reconstructed from observed time series produced by dynamical systems.

Attractors of dissipative dynamical systems (the kind of systems we are interested in) generally may have a very complicated geometry, which led researchers to call them *strange*. Since these strange attractors possess non-integer dimensions, their values can be quantified only by fractal dimensions. Non-integer dimensions are assigned to geometrical objects which exhibit an unusual kind of self-similarity and which show structure on all length scales. In general, there are five different types of fractal dimensions (Young, 1983). The simplest type, and mostl commonly known, is the capacity[5] dimension. The others are information dimension, correlation dimension, kth nearest-neighbour dimension and Lyapunov dimension. Various discussions on these dimensions and their relationships presented here can be found in Farmer et al. (1983), Young (1983), Badii and Politi (1985) and Mayer-Kress (1986).

*Capacity dimension* can be defined as follows: Suppose one covers an attractor with volume elements (such as spheres, cubes etc.) each with diameter $\varepsilon$. Let $N(\varepsilon)$ be the minimum number of volume elements needed to cover the attractor. If the attractor is a $D$-dimensional manifold—where $D$ is necessarily an integer (this is the dimension of the

[5] Here capacity is used with different meaning compared to the capacity of the learning machine discussed in the previous chapter.

dynamical system, that is, the number of state variables that are used to describe the dynamics of the system in Euclidean space)—then the number of volume elements needed to the attractor is inversely proportional to $\varepsilon^D$ (see for example York, 1983), that is,

$$N(\varepsilon)=k\varepsilon^{-D} \tag{3.34}$$

where $k$ is some constant depending on the type of volume element used. The capacity dimension $D_{cap}$ can be obtained by solving (3.34) for $D$ and taking the limit as ε approaches zero:

$$D_{cap} = \lim_{\varepsilon \to 0} \frac{\ln N(\varepsilon)}{\ln(1/\varepsilon)} \ . \tag{3.35}$$

If the limit does not exist, then $D_{cap}$ is not defined. Since a $d$-dimensional manifold locally resembles $\boldsymbol{R}^d$, $D_{cap}$ of a manifold equals the topological dimension, which is an integer.

For objects that are not manifolds, $D_{cap}$ can take on non-integer values. An interesting questions arises whether another covering (e.g., spheres instead of cubes or even a mixture of spheres and cubes) can result in a different value of $D_{cap}$. Young (1983) has demonstrated that the values of $D_{cap}$ can differ in such cases, but this results in an implication that the capacity dimension is closely related to the Hausdorff dimension, which is a measure for self-similarity of sets (see Grassberger, 1985). In such cases the minimum values of all coverings should be used.

Capacity dimension is purely a metric concept (and sometimes is termed as a box-counting dimension). It utilises no information about the time behaviour of the dynamical system. The *information dimension*, on the other hand, is defined in terms of the relative frequency of visitation of volume elements by the trajectory; thus, it is a probabilistic type of dimension. Similarly, as in the case of capacity dimension, the information dimension is defined by

$$D_I = \lim_{\varepsilon \to 0} \frac{H(\varepsilon)}{\ln(1/\varepsilon)} \tag{3.36}$$

where

$$H(\varepsilon) = -\sum_{i=1}^{N(\varepsilon)} P_i \ln P_i \ . \tag{3.37}$$

In this case the $P_i$ is the relative frequency with which a typical trajectory enters the ith volume element of the covering; see Figure 3.16.

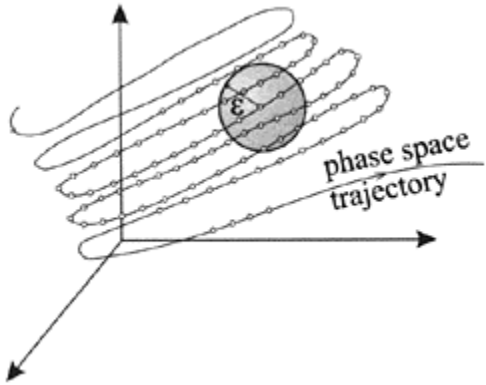



Figure 3.16. Evolution of dynamic system in phase space showing the visitation of the trajectory of a sphere in the various dimensions analysis.

One could easily recognise, from the Shannon information theory, that $H(\varepsilon)$ in (3.37) is in fact the entropy—in this case is the amount of average information needed to specify the point **x** (state of the system) with accuracy ε if the point is known to be on the attractor. This is the reason why $D_I$ is called an information dimension, which specifies how this amount of information scales with the resolution ε. For sufficiently small ε, (3.36) can be rewritten as

$$H(\varepsilon)=k\varepsilon^{-DI} \tag{3.38}$$

for some constant of proportionality *k*. In other words, the amount of information needed to specify the state of the dynamical system in a phase space increases inversely with the $D_I$ th power of ε.

Another probabilistic type of dimension, which is widely used to compute the dimension of the attractor from observables, is the *correlation dimension*. It also depends on refining a coverage of the attractor with $N(\varepsilon)$ volume elements of diameter ε, and is defined by

$$D_c = \lim_{\varepsilon \to 0} \frac{\ln \sum_{i=1}^{N(\varepsilon)} P_i^2}{\ln \varepsilon} \tag{3.39}$$

where, as before, the $P_i$ is the relative frequency with which a typical trajectory enters the ith volume element. In order to interpret the numerator of (3.39), usually for practical applications, one estimates the so-called correlation sum (or functional) from *N* points on the trajectory by

$$C(\varepsilon) = \lim_{N\to\infty} \frac{1}{N^2} \left\{ the\ number\ of\ \ points\ (x_i, x_j)\ such\ that \left| x_i - x_j \right| < \varepsilon \right\} \tag{3.40}$$

The correlation sum for a collection of points (states) $N$ in the phase space is the fraction of all possible pairs of points on the trajectory which are closer than a given distance ε in a particular norm. In the limit of an infinite amount of data ($N\to\infty$) and for small ε one can define the correlation dimension as

$$D_c = \lim_{\varepsilon\to 0} \frac{\ln C(\varepsilon)}{\ln \varepsilon} \ . \tag{3.41}$$

At this point in the discussion we set the equation (3.40) in such a form. A more detailed discussion on how correlation dimension can be estimated from time series generated and observed on some nonlinear dynamical system, is given in the next section.

The *k*th *nearest-neighbour dimension* was first formulated by Pettis et al. (1979) and is completely based on probabilistic concepts. One considers an attractor embedded in $\boldsymbol{R}^d$ with $N$ randomly chosen data points from the trajectory. If $r(k, x)$, defined as a distance between $x$ and its $k$th nearest neighbour in $\{x_i\}$, and $\bar{r}(k)$ as a mean of $r(k, x)$ are taken over $\{x_i\}$ such as

$$\bar{r}(k) = \frac{1}{N} \sum_{i=1}^{N} r(k, x_i) \tag{3.42}$$

then Pettis at al. showed that for sufficiently large $N$, there exist probabilistic functions $g$ and $c$ such that the $k$th nearest-neighbour dimension $D_{nn}$ is well-defined by the relation

$$D_{nn} = \frac{\ln k + c(x_1, \ldots, x_N)}{g(k, D_{nn}) + \ln \bar{r}(k)} \ . \tag{3.43}$$

For practical applications the estimation of $D_{nn}$ is difficult since one does not know the functions $g$ and $c$, and (3.43) is an implicit relationship.

Finally, the last dimension considered in this context can be estimated from the Lyapunov exponents (introduced earlier in the stability analysis) and is usually termed the *Lyapunov dimension*. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ be the Lyapunov exponents of an attractor of a time-continuous dynamical system. The Lyapunov dimension as defined by Kaplan and York (1979) can be written as

$$D_\lambda = j + \frac{\lambda_1 + \ldots + \lambda_j}{\left| \lambda_{j+1} \right|} \tag{3.45}$$

where $j$ is the largest integer such that $\lambda_1 + \ldots + \lambda_j \geq 0$. This means that the integer part of the dimension of the attractor is the maximal number of exponents that one can add (in descending order) such that their sum remains positive. The fractional part is found by a simple linear interpolation (the second term on the right-hand side of the above equation). If such $j$ cannot be estimated, then $D_\lambda$ is defined to be 0, meaning that in such case the

dynamical system exhibits stable equilibrium point and all Lyapunov exponents are negative. If the dynamical system has an attractor, then $\sum_{i=1}^{n} \lambda_1 < 0$ (recall the stability analysis), thus $j$ is guaranteed to be less than $n$. For an attracting limit cycle, the generic situation is that $\lambda_1=0>\lambda_2 >\ldots>\lambda_n$, and the Lyapunov dimension is 1. Similarly, the Lyapunov dimension of generic attracting $K$-periodic behaviour is $K$. If one deals with a chaotic deterministic system, the Lyapunov dimension $D_\lambda$ is almost always a non-integer. For example, in three-dimensional deterministic chaotic system (e.g. Lorenz system) with Lyapunov exponents $\lambda_+>0>\lambda\_$,

$$D_\lambda = 2 + \frac{\lambda_+}{|\lambda_-|} \tag{3.46}$$

For an attractor, $\lambda_+ + \lambda\_ < 0$ must hold, from which follows that $2<D_{\lambda<} 3$ (e.g. the dimension of the Lorenz attractor is estimated to be 2.06).

Given the different definitions and descriptions of the dimensions one could ask what are the practical implications all these dimensions? First of all, dimensions can be used to distinguish between strange and non-strange attractors, but with a careful interpretation of the obtained estimation of the dimension. One of the basic assumptions when defining the dimensions is that there is a sufficiently large number of points (states) $N$ on the trajectory. This means that in practice one should deal with quite long time series (how long we will discuss latter) which defines in a great detail the trajectory in phase space, i.e. the dynamic evolution of the system. Other issue is the influence of noise on the dimension estimation and the existence of a so-called temporal correlation in the data. All these issues will be addressed in the Section 3.3. A second practical application of the dimension is the use of it to quantify the geometrical complexity of the attractor and possibly reveal its dynamics. The dimension of the attractor (or the first integer above) gives the lower bound on the *number of essential state variables* needed to describe the dynamics on the attractor mathematically. In physical language, the dimension is a lower bound on the number of degrees of freedom of the attractor, and therefore on the number of differential equations. For example, motion on a limit cycle (dimension 1) can be described by a first-order differential equation where the variable could be the arc-length along the circle or perhaps the angle of rotation. In the case of a deterministic chaotic system, the motion of an attractor with dimension 2.6 can be modelled, at least theoretically, by a set of three differential equations (three is the first integer above the dimension 2.6).

Since this dimension only provides knowledge about the attractor, sometimes it is referred to as a local dimension. When dealing with real dynamical systems, one does not have an infinite length of the trajectory and, thus, all possible states (and dynamic regimes) of the system (one does not have full information about the attractor). This implies that the general or global dynamics of the systems may live in a higher dimension (global dimension) than the dimension of the attractor. The number of the state variables necessary to fully describe the general dynamics of the system is usually termed the *number of sufficient state variables*. The estimation of the dimensions described above also give a certain indication for this number, as presented in the Section 3.3.

## LYAPUNOV EXPONENTS

One of the most striking features for deterministic chaotic systems is the limited predictability (or unpredictability) of the future evolution of the system, despite the determinism of the system. This has been already made evident in Example 3.1 and Figure 3.1. This unpredictability is a consequence of the inherent instability of the solution, reflected by the sensitive dependence on the initial conditions. In the Section 3.2.1 we have shown that the stability of the system is closely related to the eigenvlues of the dynamical system whose generalisation is expressed by dynamic invariants known as *Lyapunov exponents*. The Lyapunov exponents are related to the average rates of divergence and/or convergence of nearby trajectories in phase space, and therefore, the measure how predictable or unpredictable the dynamical system is. In other words, they express the loss of information in time and are usually express in units of an inverse of time.

One can estimate as many different Lyapunov exponents for a dynamical system as there are phase space coordinates, i.e. principal axes, which give the average exponential rates of expansion and contraction of the attractor along these axes. Usually in practice, one is interested in the *maximal* Layapunov exponent that can be used to categorise the type of the motion of the system as presented in Table 3.1.

Table 3.1. Possible types of motion of dynamical systems and the corresponding maximal Lyapunov exponents

| Type of motion | Maximal Lyapunov exponent |
|---|---|
| stable fixed point | $\lambda<0$ |
| stable limit cycle | $\lambda=0$ |
| deterministic chaos | $0<\lambda<\infty$ |
| noise (random motion) | $\lambda=\infty$ |

From the stability analysis we have seen that a positive maximum Lyapunov exponent indicates expansion and exponential divergence of the nearby trajectories. Therefore what distinguishes strange attractors from non-chaotic attractors is the existence of a maximal positive Lyapunov exponent. However, estimating the maximal Lyapunov exponent for a dynamical system does not necessarily reveal the global dynamics of the system. In a complete data analysis one would like to determine all the Lyapunov exponents, i.e. the *Lyapunov spectrum*, which may expose additional information of the attractor of the system and its governing dynamics. One could also use the Lyapunov spectrum to compute the dimension of the attractor as discussed previously.

Following the above discussion, we now give a formal definition of Lyapunov exponents and their determination for a dynamical system described by mathematical equations. Given a continuous dynamical system in d-dimensional phase space one can monitor the evolution of a set of infinitesimal perturbations of the initial conditions in an attractor that are confined within an *d*-dimensional sphere (hypersphere), see Figure 3.17.

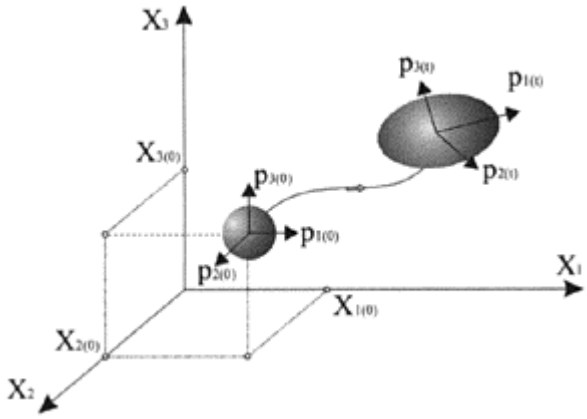



Figure 3.17. A schematic representation of the evolution of a set of initial conditions in the phase space.

Due to the locally deforming nature of the flow (effects of stretching and folding), this *d*-sphere will become a *d*-ellipsoid in time. If one orders the principal axes of this sphere (ellipsoid) from the most rapidly to the least rapidly growing, one can compute the average growth (expansion or contraction) rates $\lambda_i$ of any given principal axis $p_i$ as follows:

$$\lambda_i = \lim_{T\to\infty} \frac{1}{T} \int_0^T dt \frac{d}{dt} \ln\left(\frac{p_i(t)}{p_i(0)}\right) = \lim_{T\to\infty} \frac{1}{T} \ln\left(\frac{p_i(T)}{p_i(0)}\right), \quad i = 1,\ldots,d\,. \tag{3.47}$$

Here $p_i(0)$ is the radius of the principal axis $p_i$ at time $t$=0 (i.e. in the initial hyperspere), and $p_i(T)$ is its radius after some time $T$. The set of $\lambda_i$ is the Lyapunov spectrum. When at least one Lyapunov exponent is positive, then the dynamical system is characterised by deterministic chaos, and the initial sphere will evolve to some complex ellipsoid structure reflecting the exponential divergence of nearby trajectories (starting from very similar initial conditions) along at least one direction on the attractor. This sensitivity to small disturbances results in an inability to predict the evolution of the trajectory beyond a certain time horizon, which is approximately the inverse of the divergence rate. However, *short-time predictability exists*. When no positive Lyapunov exponent exists, then there is no exponential divergence, and thus the long-time predictability of the dynamical system is guaranteed.

A method for estimating the entire Lyapunov spectrum for nonlinear dynamical system defined mathematically by a set of differential equations (or difference equations) was already discussed in Section 3.2.1. From this section we already know that the linearised equations describe the local dynamics via the evolution of small perturbations as:

$$\dot{\mathbf{x}}' = A\mathbf{x}' \quad \text{or} \quad \mathbf{x}'(t) = e^{tA}\mathbf{x}'(0) \tag{3.48}$$

where $\mathbf{x}'(0)$ is the initial vector. Recalling the discussion in Section 3.2.1, the matrix *A* can be used to find the entire Lyapunov spectrum. In fact, the Lyapunov spectrum is a set of the eigenvalues of *A* that can be estimated at any point along the numerically integrated trajectory if the governing equations are known. In mathematical language, the Lyapunov exponent $\lambda_i$ is defined as the normalised logarithm of the modulus of the ith eigenvalue of the product of all Jacobians along the trajectory in the limit of an infinite long trajectory. In the case where the mathematical formulation of the dynamical system is not known and one deals with time series of observables (the case we are interested in), the estimation of the Lyapunov exponents uses the same concept and is be discussed in the Section 3.3.

From the facts that at least one Lyapunov exponent of a chaotic system must be positive, then one Lyapunov exponent of any limit set other than equilibrium point must be 0, and that the sum of the Lyapunov exponents of an attractor must be negative, it follows that a strange attractor must have at least three Lyapunov exponents. Therefore, *deterministic chaos can only occur in minimum three-dimensional phase space* of a dynamical system (exceptions are some maps, where chaos occurs in lower dimensions, see for example Ott, 1993). In the three-dimensional case, the only possibility for Lyapunov spectrum is (+,0,−), that is $\lambda_1>0$, $\lambda_2=0$ and $\lambda_3<0$. Since the contraction must outweigh the expansion in order to have stable three-dimensional deterministic chaos, the only possibility is that $\lambda_3<-\lambda_1$. For dynamical systems that are described by a fourth-dimensional phase space, there exist three possibilities:

(i) (+,0,−,−): $\lambda_1>0$, $\lambda_2=0$, and $\lambda_3\leq\lambda_4< 0$.
(ii) (+,+,0,−): $\lambda_1\geq\lambda_2>0$, $\lambda_3=0$, and $\lambda_4<0$. This can be the case for most real dynamical systems and has been regarded as hyper-chaos by Rössler (1979).
(iii) (+,0,0,−): $\lambda_1>0$, $\lambda_2=\lambda_3=0$, and $\lambda_4<0$. This corresponds to a chaotic two-torus. As far as reported in the literature, this case has not yet been observed.

The summary of the Lyapunov exponents for different types of attracting sets together with the qualitative description of the dynamics is presented in Table 3.2.

Table 3.2. Lyapunov spectrum of different attracting sets with the characterisation of dynamical systems.

| Approaching equilibrium state | Attracting set | | Lyapunov exponents | Dimension |
|---|---|---|---|---|
| | Flow | Poincare map | | |
| equilibrium point | point | | $0>\lambda_1\geq\lambda_2\geq\ldots\geq\lambda_n$ | 0 |
| periodic | circle | one or more points | $\lambda_1=0$ $0>\lambda_2\geq\ldots\geq\lambda_n$ | 1 |
| two-periodic | torus | one or more closed curves | $\lambda_1=\lambda_2=0$ $0>\lambda_3\geq\ldots\geq\lambda_n$ | 2 |
| *K*-periodic | *K*-torus | one or more (K-1) tori | $\lambda_1=\ldots=\lambda_K=0$ $0>\lambda_{K+1}\geq\ldots\geq\lambda_n$ | *K* |
| Deterministic chaos | strange | self similar—fractal | $\lambda_1>0$ (at least) $\sum\lambda_i<0$ | non-integer |

For example, for the Lorenz dynamical system, $\lambda_1=2.16$, $\lambda_2=0.00$, $\lambda_3=-32.4$, and $\sum\lambda<0$; thus one speaks about a chaotic deterministic dynamical system. The Lyapunov dimension is $D_\lambda=2+2.16/|-32.4|\approx 2.07$, which coincides with the correlation and capacity dimensions (see, for example, Grassberger and Procaccia, 1983).

## HOW THINGS ARE RELATED

Dimensions and Lyapunov exponents are different ways of describing and quantifying properties of the same invariant measure. All these quantities characterise aspects of the same underlying dynamics of the system. Thus, it is natural to seek for relations between them. When defining the various types of dimension of the attractor, the basic approach used is to cover an attractor with volume elements (such as spheres, cubes etc.) each with diameter ε. Some of these elements may and may not include points (states) which are available (computed or measured). If one defines the probability of finding a point in the ith n-dimensional volume element to be $P_{i\varepsilon}=M_{i\varepsilon}/N$, where $M_{i\varepsilon}$ is the number of points in the ith volume element with size ε, and $N$ is the total number of points on the trajectory, then the average probability for a given covering, $\langle P_{i\varepsilon}\rangle$, can be approximated as:

$$\langle P_{i\varepsilon}\rangle \underset{\varepsilon\to 0}{\approx} \varepsilon^D \tag{3.49}$$

This equation in fact relates the first momentum (the mean) of the variable $P_{i\varepsilon}$. One can extend this formulation to consider any moment of measure *P:*

$$\left\langle P_{i\varepsilon}^{(q-1)} \right\rangle \underset{\varepsilon \to 0}{\approx} \varepsilon^{(q-1)D_q} \tag{3.50}$$

One can extract the dimension $D$ from (3.50) as follows:

$$D_q \approx \lim_{\varepsilon \to 0} \frac{1}{q-1} \frac{\ln \left\langle P_{i\varepsilon}^{(q-1)} \right\rangle}{\ln \varepsilon} \tag{3.51}$$

For $q$=0, the value for the dimension $D_0$ can be estimated by solving the limit

$$D_0 \approx \lim_{\varepsilon \to 0} \left( - \frac{\ln \left\langle P_{i\varepsilon}^{(-1)} \right\rangle}{\ln \varepsilon} \right) \tag{3.52}$$

Since $\left\langle P_{i\varepsilon} \right\rangle$ is the average probability given by $[1/N(\varepsilon)]\sum_i P_{i\varepsilon}$, where $N(\varepsilon)$ is the number of volume elements in the covering that are not empty. Thus, $\left\langle P_{i\varepsilon} \right\rangle = 1/N(\varepsilon)$ and equation (3.52) becomes

$$D_0 \approx \lim_{\varepsilon \to 0} \left( - \frac{\ln N(\varepsilon)}{\ln \varepsilon} \right) \tag{3.53}$$

which is in fact the capacity dimension (see equation (3.35)). Similarly, for $q$=1 it can be shown that the $D_1$=$D_I$, which is the information dimension. Furthermore, for $q$=2 the equation (3.51) results in $D_2$=$D_c$, the correlation dimension. Extending this procedure to higher moments, one could speak about generalised dimensions, $D_3$, $D_4$,…, $D_n$. In general $D_0$>$D_1$>$D_2$>…>$D_n$, tgough the inequality can be replaced by an equality only in special cases (Hentschel and Procaccia, 1983). Like the various moments used in statistics to characterise the distribution of random variable, the generalised dimensions can be used to give a statistical characterisation of the multiple scaling in fractals.

We have also seen that the positive Lyapunov exponents characterise the exponential divergence of nearby trajectories. The fact that trajectories diverge directly implies a loss of information about their future position; thus the uncertainty about the future position grows with the rates of the expanding principal axes (as discussed in Figure 3.17). Pesin (1977) has found that the sum of all positive Lyapunov exponents is an upper bound of the so-called Kolmogorov-Sinai entropy (see Ruelle, 1978):

$$h_{KS} = \sum_{i:\lambda_i > 0} \lambda_i \tag{3.54}$$

Very often, Pesin's identity is the only way to obtain a good estimate of the Kolmogorov-Sinai entropy of a time series, since direct computation often requires a large amount of data (Ruelle, 1978). We address the subject of entropy estimation from time series data in the next section. Less obviously than the entropy, the dimension of the attractor is also

related to the Lyapunov exponents via the Lyapunov dimension. The relationship was first introduced by Kaplan and Yorke (1979) and is expressed by equation (3.45).

## 3.3 Reconstruction of dynamics from time series of observables

The study of mathematical nonlinear dynamical systems presented in the previous section has demonstrated that random-looking behaviour can arise even from simple nonlinear systems. Such dynamics, now termed as deterministic chaos, exhibit broadband power spectra, and complicated strange attractors possessing fractal dimensions with positive Lyapunov exponents, whose dynamics can change via bifurcation and long-term predictability cannot be guaranteed. When the mathematical formulation of the studied nonlinear system is known, than the reconstruction and identification of the dynamics using nonlinear methods is quite straightforward. However, when one deals with real-life dynamical systems (such as hydrological or meteorological systems) where one cannot observe all the variables, and furthermore one may not know completely the mathematical formulation and the total number of variables governing the dynamics, the reconstruction and identification of the dynamics becomes complicated. If one adds the fact of the inevitable presence of measurement and dynamical noise embedded into the time series of the observables, reconstruction of the dynamics of such systems becomes a real challenge.

The reconstruction of the vector space (quasi phase space) which is equivalent to the original phase space of the dynamical system from a time series is the basis of almost all nonlinear methods exploring dynamic or metric properties of the data. Having stressed the importance of the phase space for the study of dynamical systems with deterministic properties, the first important problem that we address in this section is the *phase space reconstruction* from a time series of observables, which is technically solved by *methods of time delays* (or *embedding* methods). Both, the reconstruction of the phase space using scalar and multivariate time series are described. Further, will discuss and demonstrate the necessity of finding a good embedding in order to reconstruct properly the attractor and reveal the dynamics of system. The estimation of the attractor dimension, the dimension of the phase space and the Lyapunov exponents from the time series is also discussed. Various important issues, such as the effect of the length of the time series, temporal correlations and effect of the noise on the estimation of those dynamical and geometrical invariants are also addressed. Finally, once the dynamics of the system are reconstructed, the modelling of such dynamics using both local and global models is presented.

### *3.3.1 Phase space reconstruction—method of time delay*

Most commonly, the time series obtained from dynamical system is a sequence of scalar measurements of some quantity which depends on the current state of the system, taken at multiples of a fixed sampling time:

$$s_n = s(\mathbf{x}(n\Delta t)) + \eta_n \tag{3.55}$$

Thus, we look at the dynamical system through some measurement function *s* and make observations only up to some random fluctuations $\eta_n$, the measurement noise. Let us neglect the effect of the noise at this level of presentation. The system on which the observable quantity is being measured is evolving with time. The phase space reconstruction problem is that of recreating all the states of the dynamical system when the only information available is contained in a time series, whether univariate or multivariate. A remarkable work, first started by Whitney (1936), extended by Pacard et al. (1980), and put on firm mathematical basis by Takens (1981), showed that the phase space can be reconstructed (approximated) from scalar or univariate time series of some observable *x(t)*. This is technically solved by the method of *time delay embedding*, which is known, as Takens's embedding theorem. According to this theorem, the dynamics of a time series $\{x_1, x_2,\ldots, x_N\}$ are fully captured or 'embedded' in the m-dimensional phase space ($m>d$, where $d$ is the dimension of the attractor) defined by the delay vectors

$$\boldsymbol{Y}_t=\{x_t, x_{t-\tau}, x_{t-2\tau},\ldots, x_{t-(m-1)\tau}\} \tag{3.55}$$

where $\tau$ is suitable *time delay* and $m$ is referred to as an *embedding dimension*. Let us stress a few important considerations about the definition of the embedding dimension. When an attractor of dynamical system exists, its dimension (be it integer or fractal) is smaller than the dimension of the phase space. One can take advantage of this (in a modelling sense) and try to develop a model of the lower-dimensional dynamical system that describes only the motion of the attractor (since it is defined by available data describing the visitation of the trajectory). This can be achieved by embedding the attractor in a smooth manifold (smooth in a sense of a non-intersecting trajectory) and restricting our model of the dynamics only to this manifold. Recall that the *k*-dimensional manifold is a geometrical model, i.e. set of points that locally resembles $\boldsymbol{R}^k$. More precisely, $M$ is a *k*-dimensional manifold if for each point $x \in M$, there exist an open neighborhood of $x$ such that this neighborhood is diffeomorphic (a smooth mapping exist) to some other open neighborhood in $\boldsymbol{R}^k$. The lowest possible dimension of such manifold is called an embedding dimension and, thus, gives the number of essential variables to model the dynamics of the system (as discussed previously). Attractors that are topological structures (points, limit cycles and tori) are submanifolds of the manifold in which they are embedded. Strange attractors (ones that have fractal dimensions) are *not* submanifolds. When we try to reconstruct the attractor from a time series of observables, the dimensionality of the manifold that embeds it, it is not known *a proiri*. Thus, one has to search for a proper embedding dimension, such that the structure of the attractor becomes invariant. According to Whitney (1936), any smooth manifold of dimension $d$ can be smoothly embedded in $m=2d+1$ dimension. Taken's theorem (1981) shows that if the dimension of the manifold containing the attractor is d, then embedding the data in a phase space with dimension $m\geq 2d+1$ preserves the topological properties of the attractor. Sauer et al. (1991) further discussed the generalisation of the embedding theorem, emphasising the importance of the fractal dimension of the attractor for estimation of the minimal dimension of the embedding space, i.e. $m>2d$. Some authors (see, for example, Abrabanel et al., 1991) suggest that, in practice, $m>d$ is sufficient.

The above discussion shows that the main result of the various variations of the embedding theorem is that it is not the dimension of the underlying true phase space that

is important for the necessary embedding of the time series, but the fractal (or integer) dimension of the attractor *d*. In natural dissipative dynamical systems this dimension can be much smaller than the dimension of the true phase space of the system. In fact, low-dimensional dynamics have been observed in various complex dynamical systems, including hydrological, meteorological and oceanographic systems (we discuss these in detail in Chaper 6, which deals with the applications). However, one should not forget that the search for the dimension of the true phase space of the system from time series is equally important, since it may enhance our knowledge and understanding of the underlying dynamics and reveal further the number of the sufficient variables necessary to fully describe the motion of the system mathematically.

Apart from the Taken's time delay embedding method for the reconstruction of the phase space that is most commonly used, several other methods exist that may be suitable for particular application and better representation of the data, especially if the data are noisy and one wants to reduce the noise level implicitly. One of the embedding methods which is closely naturally related to the mathematical description of the nonlinear dynamical systems is the so-called *derivative coordinates*. In this case, the phase space coordinates are constructed from the derivatives of the observable $\{s_t, \dot{s}_t, \ddot{s}_t, \ldots\}$. Numerically, one should form the adequate differences between successive observations, i.e. $\ddot{s}_t \approx (s(t+\Delta t)+s(t-\Delta t)-2s(t))/\Delta t^2$ $\dot{s}_t \approx (s(t+\Delta t)-s(t-\Delta t))/2\Delta t$, etc. Then the state vector of the system at time t is defined by $\boldsymbol{Y}_t = \{s_t, \dot{s}_t, \ddot{s}_t, \ldots\}$. One could also use integrals instead of derivatives or mixtured representation, based on the physical problem analysed. The advantage of phase space reconstruction using derivative coordinates is their clear physical meaning. However, their drawback lies in their sensitivity to noise. In order to illustrate this, let us assume that the measurement noise $\eta(t)$ is identically distributed with variance $\sigma^2_{noise}$, zero mean, and normalised autocorrelation function $c_{\text{noise}}(\tau)$. Further let the observable be a recorder with a high sampling rate $(1/\Delta t)$ such that the successive observations are strongly correlated (with $\sigma^2_{obs} c_{\text{obs}}(\tau)$). One can claim that the first derivative is corrupted by a larger noise level than the original signal itself. If $x(t)$ is the "clean" variable and $s(t)=x(t)+\eta(t)$ is the observed signal, then the first derivative can be written as

$$\dot{s}(t) = \frac{1}{2\Delta t}\left(s(t+\Delta t)-s(t-\Delta t)\right) = \frac{1}{2\Delta t}\left(x(t+\Delta t)+\eta(t+\Delta t)-x(t-\Delta t)-\eta(t+\Delta t)\right) \tag{3.56}$$

The variance of the derivative, which now becomes $(x(t+\Delta t)-x(t-\Delta t))/2\Delta t$, is then $\sigma^2_{obs}(1-c_{obs}(2\Delta t))/\Delta t$, and the variance of the noise $\sigma^2_{noise}(1-c_{noise}(2\Delta t))/\Delta t$. Therefore, the relative noise level of the first darivative $\dot{s}(t)$ in root mean square sense is

$$noise\ level_{\dot{s}(t)} = \frac{\sigma_{noise}}{\sigma_{obs}}\sqrt{\frac{1-c_{noise}(2\Delta t)}{1-c_{obs}(2\Delta t)}} \tag{3.57}$$

which can be much larger than the relative noise level of the original signal $\sigma_{noise}/\sigma_{obs}$, if the autocorrelation of the signal decays considerably slower than the autocorrelation of the noise: a real situation which occurs in most flow-like dynamical systems (runoff, water levels etc.). Analogous considerations can be made of the higher derivatives, which even further amplify the noise level. Therefore, derivative techniques have to be used with care, and usually may require additional data preprocessing using nonlinear low-pass filtering techniques, such as *continuous wavlet transform*.

Another kind of data analysis technique that is being used for reconstruction of the phase space of dynamical systems from observables is the so-called *singular value decomposition technique* (Broomhead and King, 1986). This technique has appeared in the literature under various different names, such as temporal principal component analysis (PCA), singular spectrum analysis (SSA), Karhunen-Loeve transformation, or empirical orthogonal functions (EOFs) method. However, the basic idea is to characterise the time series by its most relevant components in a delay embedding space $\boldsymbol{R}^M$ (we do not use the term phase space in this context) where $M$ is most probably too large. This technique uses a set of all delay vectors $x_i=x(t_0-M\Delta t)$, $1\le M\le N$ (using time delay equal to $\Delta t$) and estimates the eigenvalues $\lambda_n$ and eigenvectors $\rho_n$ (which are orthogonal) of their $M\times M$ *covariance* matrix $C$:

$$C_{i,j} = \frac{1}{N-M+1}\sum_{n=1}^{N-M+1} x_{n-M+i}x_{n-M+j} \quad , \quad i,j = 0,\ldots,M-1 \tag{3.58}$$

The set of the delay vectors form an irregular cloud in $\boldsymbol{R}^M$. Often, there are directions in which the cloud extends, or does not extend. The eigenvalues of the covariance matrix are squared lengths of the semi-axes of the hyper-ellipsoid which best fit the cloud of data points, and the corresponding eigenvectors give the directions of those axes. The eigenvalues describe variables that are statistically linearly independent, while the eigenvectors span the embedding space, which is sometimes called the singular space. The most relevant directions in this space are thus given by the vectors corresponding to the largest eigenvalues, which can be used in further analysis and transformations. If there are very small eigenvalues, the corresponding directions may be neglected and considered as noise level of the observable. In order to illustrate this, the eigenvalue spectrum computed on the *y(t)* variable of the Lorenz system is presented in Figure 3.18a. Figure 3.18b shows both time series: the original and reconstructed using the first 10 components, i.e. EOFs. Figure 3.19 presents a two dimensional projections of the original and reconstructed attractor using the reconstructed variable *y(t)*.

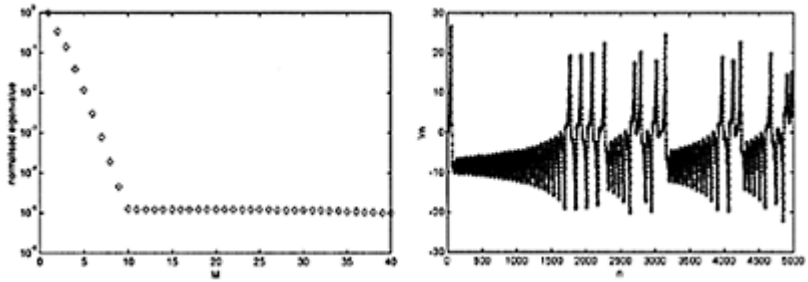


Figure 3.18. (a) The eigenvalue spectrum of the covariance matrix of the *y(t)* variable of Lorenz system. The first 10 eigenvalues and the corresponding eigenvectors are more relevant. The “noise floor” can clearly be seen; (b) Original (dots) and reconstructed (line) time series of the variable *y(t)*. Although the root mean squared error RMSE is 0.0014, the plot of the differences between the two time series shows the presence of a large error in the transition region of the trajectory from one to the other wing of the attractor, due to the nonlinear character of this transition.

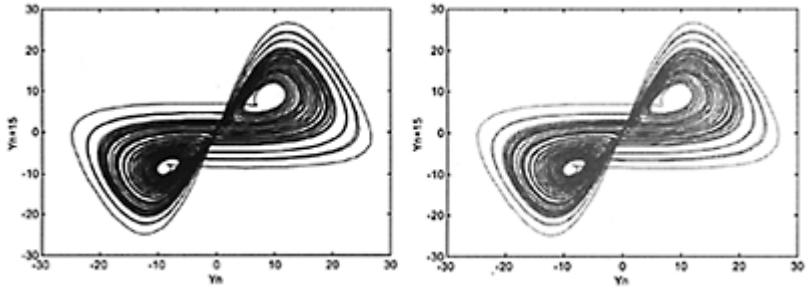


Figure 3.19. Two-dimensional projection of the attractor of the Lorenz system (a) original times series; and (b) reconstructed time series using the first 10 components.

The main drawback of this reconstruction approach is that it ensures linear independence of the variables. In real practical applications this might not be the desired result, since the nonlinearity of the system is what we are interested in. Therefore, this approach can be very useful in quantifying the nonlinearity present in the dynamics of the system. One could create surrogate time series (as we did in the above example) using linear independent components and further apply appropriate statistical tests to the residuals in order to investigate the presence of nonlinearity in the data (see, for example, the BSD nonlinearity test proposed by Brock at al., 1987).

Which method is appropriate for the reconstruction of the phase space of dynamical system using time series data in general depends on the type of the application and dynamical system being analysed, and the quality of the available data. Frazer (1989) demonstrated by several examples that Taken's time delay method, with appropriately chosen time delay $\tau$ and embedding dimension $m$, is superior to the singular value decomposition method and derivative coordinates. In general, we do not favour or discard any of these methods, but in real applications try to exploit them all, since we argue that the nonlinear time series analysis and modelling is an interactive process. Note also that up to this point of discussion, we have not presented the embedding theorems in a rigorous mathematical language; instead we have focused on explaining them in physicists' words. The mathematical aspects of the embedding theorems are well described in the original "embedology" work by Sauer et al. (1993) and the paper by Takens (1981).

### *3.3.2 Estimating dimensions from time series*

The importance of estimating various dimensions for the proper reconstruction of the phase space from time series of observables has been already highlighted. The various ways to characterise the self-similarity of a geometrical object (such as the attractor) have also been discussed in the previous section. However, there still remains a question about how to estimate the fractal dimension from time series, which are usually limited in their

length and polluted by noise. The most widely used fractal dimension quantifier is the *correlation dimension* $d_c$, which is based on the correlation integral or function analysis (Grassberger and Procaccia, 1983a,b). Obtaining a noninteger, finite $d_c$ for a time series when a corresponding stochastic surrogate does not exist demonstrates fractal scaling and indicates possible chaotic dynamics. This algorithm uses the phase space reconstruction from a scalar time series using the method of delays (3.55), where the reconstruction procedure involves the choice of time delay τ. The correlation sum *C(r)* for a collection of points $\boldsymbol{Y}_t$ in some vector space is the fraction of all possible pairs of points which are closer than a given distance *r* in a particular norm; see Figure 3.20a.

$$C(r) = \frac{1}{N_{ref}} \sum_{i=1}^{N_{ref}} \frac{1}{N} \sum_{j=1}^{N} H\left(r - \left|\boldsymbol{Y}_i - \boldsymbol{Y}_j\right|\right) \tag{3.59}$$

where *H* is the Heaviside step function, *H(y)=1* for *y*>0 and *H(y)=0* for *y*≤0, *r* is the radius of the sphere centered on $\boldsymbol{Y}_i$, *N* is the number of points in $\boldsymbol{Y}_t$, and $N_{ref}$ is a calibrated number of reference points taken from $\boldsymbol{Y}_t$ that are needed to yield consistent statistics. The norm $|\boldsymbol{Y}_i - \boldsymbol{Y}_i|$ is the standard Euclidean norm. The sum just counts the pairs ($\boldsymbol{Y}_i$, $\boldsymbol{Y}_j$) whose distance is smaller than *r* or, put in other words, the relative frequency with which a typical trajectory enters the ith volume element (sphere). The correlation function *C(r)* is estimated for the range of *r* available from the time series and for several embedding dimensions *m*. Then *C(m,r)* is inspected for the signatures of self-similarity, usually by estimating the slope of *Log C(r)* versus *Log r* plot. If the time series is characterised by an attractor, then for positive values of *r*, the correlation integral *C(r)* is scaled to the radius *r* by the power law:

$$C(r) \cong \alpha\, r^{\upsilon} \tag{3.60}$$

where υ is called the correlation exponent (slope of the *Log C(r)* versus *Log r* plot) and a is a constant. The slope can be generally estimated by the least-squares fit of a straight line over a certain range (length scales) of r, known as the *scaling region* (see Figure 3.21 for an illustration).

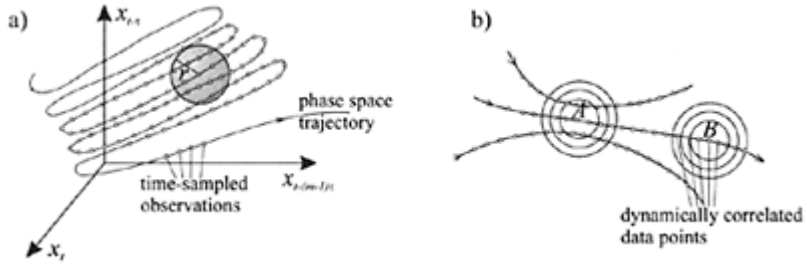



Figure 3.20. (a) Evolution of dynamic system in phase space showing the time-sampled data points and the

neighborhood of the sphere in the correlation integral analysis (b) Influence of the temporal correlation on correlation integral analysis. While for point *A* there are some dynamically uncorrected neighboring points (lying on different trajectories), all neighboring points for point *B* are temporally correlated and thus stimulate a correlation dimension close to 1.

For a random process, υ varies linearly with increasing of *m*, without reaching a saturation value, whereas for deterministic process, the value of the correlation exponent υ saturates and becomes independent of *m* for increasing embedded dimension. The saturation value $d_c$ is defined as the correlation dimension of the attractor of the time series. If the correlation dimension $d_c$ leads to a finite integer value, the underlying dynamics of the system is considered to be dominated by a strong periodic determinism. If the value of $d_c$ is fractal and usually small then the system is thought of as being dominated by a low-dimensional deterministic chaotic dynamics governed by the geometrical and dynamical properties of an attractor.

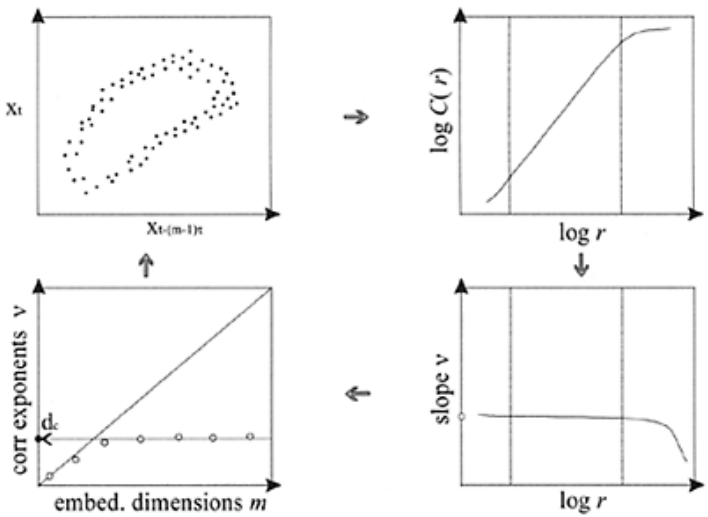



Figure 3.21. Graphical illustration of the procedure for estimating the correlation dimension.

As we already pointed out, the correlation dimension of the attractor indicates the dimension of the phase space ($m$=2$d$+1) required for a smooth embedding the attractor. This in turn provides information on the number of essential variables necessary to describe the dynamic evolution of the system. The embedding dimension where the correlation dimension reaches its saturation value provides the upper bound on $m$, and it is sufficient to fully describe the dynamics of the system.

Soon after its publication in 1983 the correlation dimension became a quite popular dimension estimation tool and since then many researchers have reported evidence of the existence of low-dimensional attractors for a vast number of dynamic systems, including weather and rainfall (e.g. Nicolis and Nicolis, 1984; Fraedrich, 1986; Esex et al., 1986; Tsonis and Elsner, 1988; Hense, 1987; Rodriguez-Iturbe et al., 1989; Sharifi *et al.*, 1990; Tsonis *et al.*, 1993; Jayawardena and Lai, 1994; Georgakakos *et al.*, 1995; Koutsoyiannis and Pachakis, 1996; Sivakumar *et al.*, 1998, 1999; Sivakumar, 2000). Usually, these conclusions were made whenever the authors succeeded in fitting a straight line to a portion of the *Log C(r)* versus *Log r* plot. Furthermore, several authors found the existence of correlation dimensions on nondeterministic data sets (e.g. Theiler, 1986 and 1991; Osborne *et al.*, 1986; Osborne and Provenzale, 1989), which seemed to be in contradiction with the fact that stochastic data are of infinite dimension. Closer examination by several authors (see Sivakumar, 2000 for overview) showed that the straightforward application of the Grassberger-Procaccia correlation dimension suffers for several problems, such as the number of the points needed for reliable estimation of the correlation dimension and the choice of appropriate time delay for the reconstruction of the phase space (this will be addressed in the text below).

An important consideration for assessing the reliability of the correlation dimension of the attractor is the size of the embedded time series. For a finite data set one can argue that there are parts of the scaling region $r$ below which there are no pairs of points (depopulation). At the other extreme, when the radius $r$ approaches the diameter of the cloud of points in phase space, the number of pairs of points no longer increases as the radius increases. In geometrical terms, the time series must be long enough to contain the points along the "edges" of the attractor. This lack of points on lower and higher length scales results in a s-shaped *Log C(r)* versus *Log r* plot, thus, requiring careful interpretation of these curves and involving a nontrivial extrapolation from a finite data set to a evidently existing attractor. Several authors studied the necessary length of the time series and the *number of points* needed to reliably estimate the correlation dimension of the attractor. Wilcox et al., (1991) and Tsonis et al., (1993) suggested criteria such as $10^A$ or $10^{(2+0.4m)}$ data points, where $A$ is the greatest integer lower than $d_c$ and $m$ the embedding dimension. These criteria imply, for example, that to investigate the existence of an 5-dimensional attractor using some hydrological observable, one needs as many as 10,000 points, which requires 27 years of daily hydrological records. Also, different variables for a different dynamical systems may require a different number of points to obtain the correlation dimension, depending on how each is coupled to the rest of variables and whether each exhibits thresholds in its behaviour (see, for example, Islam et al., 1993). Other authors (e.g. Rodrigues-Iturbe et al., 1989) suggested continuating to decrease the total available length of the time series and estimating the correlation dimension until significant changes in the results are observed in order to obtain the

minimum number of necessary data points (termed in (3.59) as the number of reference points).

Another important consideration while estimating the correlation dimension from a time series is the *effect of noise*. If the data are noisy, then below a length scale of a few multiplies of the noise level the method detects the fact that the data points are not confined to the fractal structure of the attractor, but are scattered over the whole available phase space. Thus, the local correlation exponents υ increase and at the noise level they reach the value of the embedding dimension *m*. Some authors (e.g. Kanz et al., 1993) recommend preprocessing the noisy time series using nonlinear noise reduction methods (to be discussed), before passing the data to the correlation dimension analysis.

Finally, one of the major problems of the dimension estimation from time series data is the problem of *temporal correlations* (see Figure 3.20b), which was not properly addressed by the authors seeking a low-dimensional attractor in hydrological systems. The most important temporal correlations are caused by the fact that data close in time are also close in space, a fact which is not only true for purely deterministic systems but also for many stochastically driven processes. Rather than this continuity in time, correlation dimension analysis look into the smoothness in the phase space, implying that similar present states evolve into similar states in near future, thus providing a measure of the static geometrical properties of a possible fractal attractor. Theiler (1986) studied the problem of temporal correlations and proposed a simple modification to the correlation function analysis (3.59) in order to exclude those points which are temporally correlated. This technically means that the second sum in the correlation function (3.59) is started after a typical correlation time $t_{\min}=n_{\min}\Delta t$ has elapsed,

$$C(r)=\frac{1}{N_{ref}}\sum_{i=1}^{N_{ref}}\frac{1}{N-n_{\min}}\sum_{j=n_{\min}}^{N}H\left(r-\left|\mathbf{Y}_i-\mathbf{Y}_j\right|\right) \tag{3.61}$$

The detection of the temporal correlations and the determination of a safe value of the correlation time $t_{\min}$ has been solved by Provenzale et al., (1992) by introducing the so-called *space time separation plot*. The idea is that in the presence of temporal correlations the probability that a given pair of points has a distance smaller than *r* does not only depend on *r* but also on the time that has elapsed between two measurements. This dependance can be detected by plotting the number of pairs as a function of two variables: the time separation $\Delta t$ and the distance *r*. In this manner, one can obtain contour lines with the same probability as a function of $\Delta t$ and *r*, and thus identify the correlation time $t_{\min}$. The correlation time $t_{\min}$ for a flow-type of dynamical system with a high sampling frequency can be quite large (up to 500 samples), which in turn reduces the total number of points for the estimation of the correlation function (3.61). If one deals with a long time series, this statistical loss is marginal. The effect of the temporal correlations on the dimension estimation for hourly, daily and weekly rainfall data for De Bilt meteo station in the Netherlands was studied by Velickov (2001). It was shown that if the temporal correlations are not properly encountered in the dimension estimation, one could severely underestimate the correlation dimension of the attractor, and thus the embedded dimension of the reconstructed phase space of the rainfall dynamics.

The correlation and Lyapunov dimensions, which are mostly used to estimate the attractor dimension from time series, are one way to estimate the optimal embedded

dimension *m*. Another way to estimate the optimal value of *m* is to look for *false nearest neighbours* (FNN) in phase space at a given value of *m*. Consider a situation that an $m_0$—dimensional delay reconstruction is the embedding, but an ($m_0$−1)—dimensional is not. The question is what happens when passing from $m_0$ to $m_0$−1 (note the similarity with a Poincarè section). One simply projects the along one coordinate and thus maps different parts of the attractor onto each other. When selecting a number of close points from such a region of the $R^{m_0-1}$, the images of the points will form different groups, depending from which part of the attractor the points are sampled. This lack of a unique location of all the images in $m_0$−1 dimensions is reflected by finding false neighbours, meaning that the determinism is violated. When increasing *m*, starting from small values of the embedding, one can thus detect the minimal (optimal) embedding dimension by finding no more false neighbours. This FFN method for estimating the optimal embedding dimension was first proposed by Čenus & Pyragas (1988) and further elaborated by Kennel et al. (1992). It was found by Kennel et al. (1992) that for noise-free time series the percentage of false neighbours will drop to zero when the optimal embedding dimension *m* is reached. A further increase in the embedding dimension will not affect the false neighbours since the attractor will be properly unfolded. In a presence of noise, one should not expect a drop in the percentage of false neighbours to zero in any dimension. Furthermore, if the time series in question is stochastic, there will not be a substantial drop of the false neighbours with the increase of the embedding dimension.

In order to illustrate the dimension estimation from time series observed on a real dynamical system, we briefly present here some of the results obtained by analysis of 10min water levels (328608 data points) observed at the Hoek van Holland tidal station in the Netherlands (a detailed analysis and discussion is presented in Chapter 6). Figure 3.22 show the correlation integral for the water level data at different length scales.

From Figure 3.24 one can see the saturation value of the correlation exponent for properly chosen time delay for the embedding of the water level time series (the optimal time delay is $\tau$=21 in this case). This indicates the importance of finding the optimal time delay in order to properly unfold the attractor (if one exists) in the phase space. The value of the correlation dimension of the attractor in this case is estimated to be $d_c$=2.40. Taking into account the previous discussion about the estimation of the embedding dimension *m*, if one uses Taken's embedding theorem the embedded dimension (integer number) of the manifold which contains the attractor is about *m*=6. If one uses Withney's recommendation, the embedding dimension is about *m*=5. Abrabanel's recommendation (the first integer above the correlation dimension) leads us to *m*=3. The false nearest neighbours method gives an estimation of the embedding dimension *m*=6; see Figure 3.25.

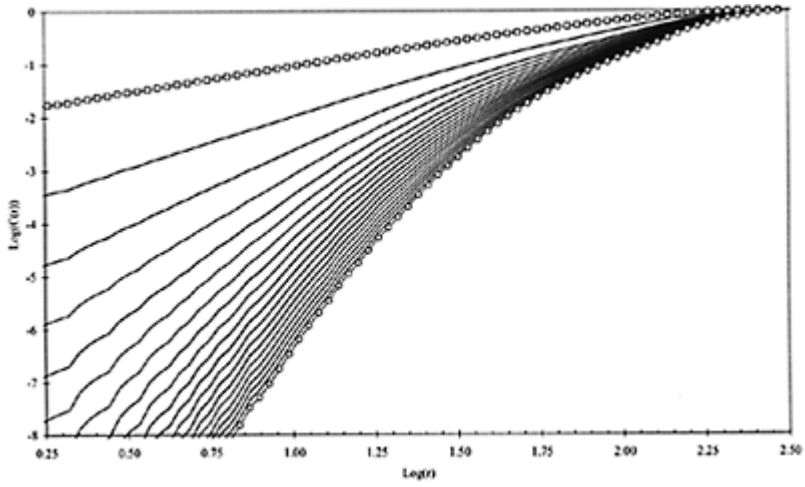



Figure 3.23. Correlation integral (sum) for the Hoek van Holland water level data (period 1990–1996, 10min data). Double logarithmic plot was chosen for better visual presentation of the power law scaling between the correlation sum *C(r)* and the length scales r. The correlation sum was computed for different embedding dimensions (the line with squares corresponds to embedding dimension 2 and the line with open circles correspond to embedding dimension 20). After embedding dimension *m*=12 the lines become parallel and thus the slope (correlation exponent) saturates, see next Figure 3.24.

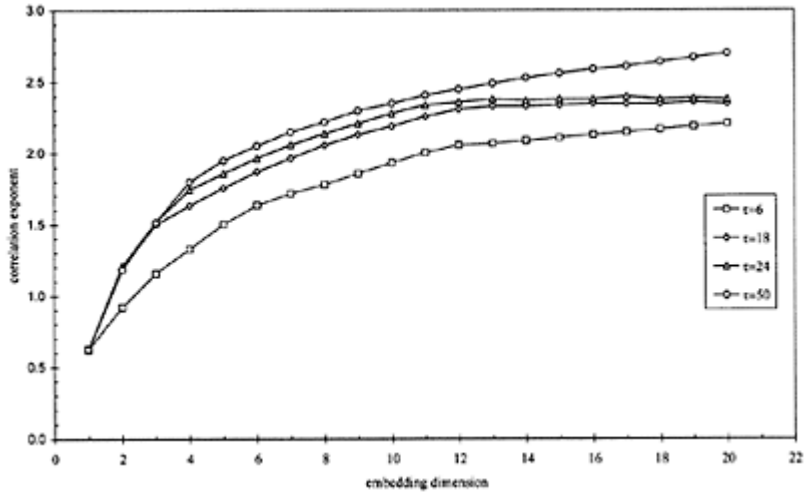



Figure 3.24. Relationship between the correlation exponent v and embedding dimension *m* for the Hoek van Holland 10min interval water level data using different time delays τ. The correlation exponent increases with an increase of the embedded dimension up to a certain value and further saturates (when using time delays between τ=18 and τ=24). The saturation value of the correlation exponent, that is the correlation dimension, is 2.40 (uncertainty 0.5) which indicates presence of an attractor in the dynamical system.

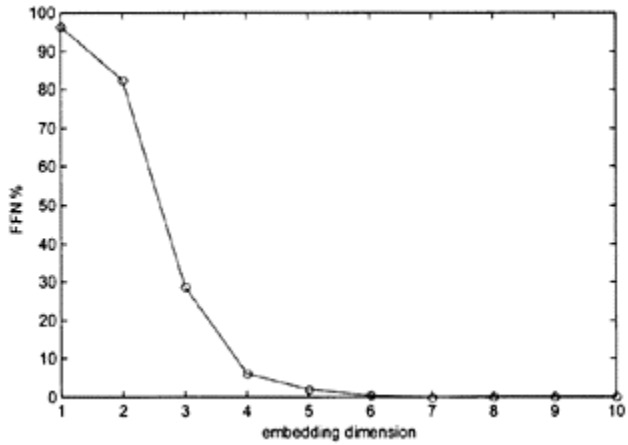



Figure 3.25. The percentage of the false nearest neighbours as a function of the embedding dimension for the water level data at Hoek van Holland tidal station.

This figure shows that the percentage of the FFN drops to about 1% with the embedding dimension $m$=6, and remains unchanged with a further increase in the embedding dimension. The Lyapunov dimension estimated on the basis of the Layapunov exponents for the same data set is $d_\lambda$=5.55, (see next Section 3.3.4), thus indicating an embedding dimension of $m$=6. As previously pointed out, this embedded dimension reveals the *essential dimension* of the phase space (and the number of the essential variables) necessary to model the dynamics of the attractor. The *sufficient dimension* of the phase space, necessary to fully describe the global dynamics of the system, can also be identified, for example from Figure 3.24, as a dimension where the correlation exponent reaches its saturation value (12 in this case). More discussion on the dimension estimation is presented in Chapter 6. From the brief presentation of some of the results it is obvious that there is no real "recipe" prescribed for the estimation of various geometrical and dynamical quantities based on the time series. Extracting and modelling the dynamics of dynamical systems from time series obviously requires a highly interactive approach and a careful analysis of the obtained results.

### *3.3.3 Finding appropriate time delay*

The time delay $\tau$ between successive elements in the delay vectors (3.55) is not a mathematical subject of the embedding theorems, since they consider data with infinite precision. Embeddings with the same $m$ but different $\tau$ are equivalent in the mathematical sense, but in real applications, the delay time $\tau$ needs to be appropriately chosen in order to fully capture the structure of the attractor. If $\tau$ is too small then the delay vectors are not independent, such that all points are accumulated around the bisectrix of the

embedding space, resulting in a loss of the characteristics of the attractor structure. If $\tau$ is very large (i.e. much larger than the decorrelation time of the system), the different coordinates (delay vectors) may be almost dynamically uncorrelated. In this case the reconstructed attractor may become very complicated, even if the underlying 'true' attractor is simple. Casdagli *et al.* (1991) discussed rather rigorously the influence of the choice of $\tau$ on the quality of the phase space reconstruction, but did not come up with a real practical method for determining an optimal value of $\tau$. In the literature, the issue of how to estimate $\tau$ has been emphasised greatly, and at east a dozen different methods have been suggested. For example, the straightforward choice of $\tau$ is usually made with the help of the zero-crossing autocorrelation function. Tsonis and Elsener (1988) suggested that the time delay may be chosen as the lag time at which the autocorrelation function falls below a threshold value which is commonly defined as $1/e$, specially if the autocorrelation function exhibits an exponential decay. If the data are suspected to be very noisy, $\tau$ has to be larger than the time when the normalised autocorrelation function decays to $1-\sigma^2_{noise}/\sigma^2_{signal}$. However, it must be pointed out that the autocorrelation function exploits the linear structures in the data.

In the terms of nonlinear methods, the choice of $\tau$ corresponding to the first minimum of the time delayed *mutual information* (Fraser and Swinney, 1986) demonstrated good performances in the practical applications. This delayed mutual information is based on the Shnanon's entropy and can be computed as follows: Given a time series of observable $s$, one can calculate the transitional probabilities $P_s(s_i)$ that a measurement $s$ yields $s_i$. The information entropy is thus defined as:

$$H(s) = -\sum_{i=1}^{N} P_s(s_i) \log P_s(s_i) \ . \tag{3.62}$$

The entropy expressed in (3.62) is a measure of the uncertainty associated with the measurement $s$. In other words, one can think of the degree of surprise when one reads the value of the measurement $s$. Low-probability (unexpected) measurements carry greater entropy than the high-probability (expected) measurements. The question now is how the value of the measurement $x(t+\tau)$ depends on $x(t)$ as a function of the time delay $\tau$. If one denotes $s=x(t)$ and $q=x(t+\tau)$, then the conditional entropy can be written as:

$$H(q,s_i) = -\sum_{j=1}^{N}\left(\frac{P_{sq}(s_i,q_j)}{P_s(s_i)}\right)\log\left(\frac{P_{sq}(s_i,q_j)}{P_s(s_i)}\right) \tag{3.63}$$

where $P_{sq}(s_i, q_j)$ is the probability that measurements of $s$ and $q$ yield $s_i$ and $q_j$. In this case one could define $H(q, s_i)$ as the uncertainty of $q$ given $s_i$. The mutual information is then defined as the amount by which a measurement of $s=s_i$ reduces the uncertainty of $q$:

$$I(q, s_i)=H(s_i)+H(q)-H(q, s_i). \tag{3.64}$$

If the time delay is chosen to coincide with the first minimum of the mutual information, than the reconstructed state vector $\mathbf{Y}_t$ will consist of delay components that possess minimal mutual information between them. The mutual information method is probably

the most comprehensive method of determining proper time delays when reconstructing the dynamics of the systems from time observables. The only drawbacks of this method are that it requires a large amount of data and that it is computationally expensive. In order to illustrate the difference in the proper time delay determination, we will again consider the Lorenz dynamical system. Figure 3.26 shows the auto correlation function and the mutual information for the variable *y(t)*.

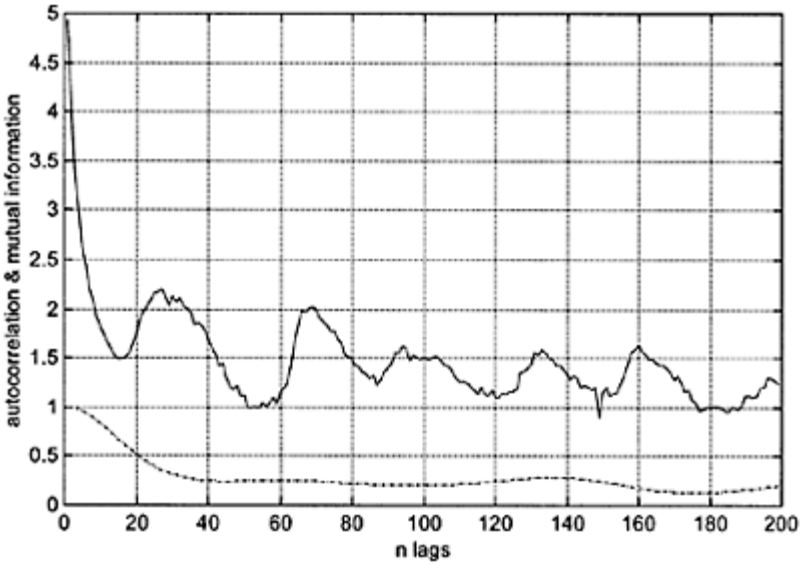



Figure 3.26. The autocorrelation function (dash-dotted line) and the mutual information (solid line) as a function of time lags for the *y(t)* variable of the Lorenz system.

According to Figure 3.26, the autocorrelation function decays smoothly (due to the flow-type nature of the Lorenz dynamical system) and thus does not even exhibit a zero crossing for the first 300 time lags (a zero crossing exist at a time lag of 580, but is not presented on this figure for better representation). On the other hand, the mutual information reaches its minimum at time lag of $\tau$=18. It is well-know for the Lorenz system that the optimal time delay is ¼ of the mean orbital period, which in this case correspond to $\tau$=20. Thus the mutual information is able to properly determine the optimal time lag in comparison with the autocorrelation function for this particular example. The time delay as the lag time at which autocorrelation function falls below a threshold value of $1/e$=0.368 corresponds in this case to $\tau$=28, which provides a better estimate than the zero-crossing criterion. The reason for this is that a typical trajectory of the Lorenz system stays some time on one wing of the attractor (as discussed previously), spiraling from the inside to its border before it jumps to the other wing. A record of the *y*-

variable thus shows alternations of oscillations around a negative mean close to zero, which reflect a smooth decay of the autocorrelation function and do not indicate that the average period of the motion on a single wing is of relevance.

The autocorrelation function and the mutual information as a function of the time lags for the water level data (10min interval) at Hoek van Holland tidal station (1990–1996) is presented in Figure 3.27. Both functions suggests a similar optimal value for the time delay of $\tau$=20 time steps, which corresponds to 3.33 hours.

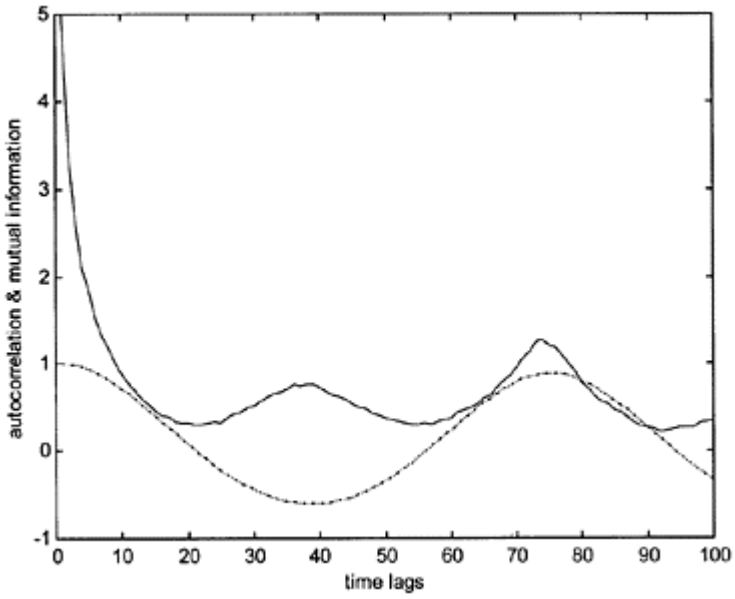



Figure 3.27. The autocorrelation function (dash-dotted line) and the mutual information (solid line) as a function of time lags for the hourly water level time series at Hoek van Holland tidal station.

Another alternative for estimating the time delay reported in the literature is to build 'local' prediction models (Farmer and Sidorovic, 1987) of the attractor dynamics utilising different values for $\tau$, while trying to minimise the prediction error. Any global optimisation method can be used to search for optimal value of $\tau$. However, this method is very sensitive to noisy data. Since all of these methods yield different values for the time delay, we advocate that in a particular application one should perform a sensitivity analysis and estimate the dynamic and metric invariants by varying $\tau$, and thus in a way optimise the time delay (see Velickov, 2001 for discussion).

### *3.3.4 Estimating Lyapunov exponents form time series*

In Section 3.2.4 we discussed how to obtain the complete Lyapunov exponent spectrum when one knows the mathematical formulation of the dynamical system, that is, the mapping from one state to another state of the system in phase space is known. Same analogy has resulted in different algorithms for estimating the Lyapunov exponents from observables. Following the definition of the Lyapunov exponents, in order to estimate the exponents one simply has to consider nearby points on the attractor and monitor their long-term evolution. Wolf et al. (1985) first presented an approach by which the largest Lyapunov exponent, $\lambda_1$, is estimated once the attractor has been reconstructed from the time series. Theoretically, $\lambda_1$ is estimated by monitoring the long-term evolution of a pair of nearby orbits. However, one has to point out that the reconstructed attractor from the time series contains just one trajectory. The reconstruction can, nevertheless, provide points that may be considered to lie on different trajectories if one chooses two points whose temporal separation in the original time series is at least one mean orbital period of the dynamics of the system. As long as their spatial separation in the reconstructed attractor is small, those two points can be considered to define the early state of the first principal axis. By monitoring their separation, and when it becomes large two new points can be sampled. Repeating this procedure many times from using different pairs of points gives the average estimate of the largest Lyapunov exponent $\lambda_1$. The algorithm of Wolf et al. (1985) was further modified and improved by Frank (1990) for estimation of $\lambda_1$ in case of noisy data sets.

In order to estimate the whole Lyapunov spectrum, according to the Jacobian method (see, for example, Eckman et al., 1986) a neighbourhood of $l$ points within a small distance is considered around a reference point on the reconstructed trajectory. Then a local linear map that maps the whole neighbourhood into a neighbourhood after some time horizon $T$ is derived. To obtain the mapping, the location of the neighbours at each time step is monitored. For sufficiently small neighbourhoods and time intervals, the evolution of the nearby states is approximated by equation $\mathbf{x}'_{n+1} = A\mathbf{x}'_n$ (recall the stability analysis in Section 3.2.2). Therefore, information about the *local phase space expansions and contraction* rates is contained in the linearised equations, which provides the reasoning behind obtaining *local linear maps*, though the general mapping of the neighbourhoods in the attractor is nonlinear. When one tries to obtain a mapping from the reconstructed attractor, the basic assumption is thus that the $l$ points are small fluctuations from the reference point on the trajectory and that the evolution of each fluctuation obeys a local linear law. For each point a value of the local mapping parameter $a$ is obtained. For many points the idea is to find an optimal set of parameters that minimises the linear regression error from all neighbours. Extending the analogy to $n$ dimensions results in the task of estimating an $n \times n$ matrix $A$ whose eigenvalues provide the Lyapunov exponents.

Let us illustrate this approach on a simple example. For simplicity, we assume that the time series of some observable is *{s(t)}:*3,1,2,1,1,3,2,3,3,5,3,3,2,1. Furthermore, a three-dimensional phase space ($m$=3) is reconstructed using dime delay $\tau$=1; see Table 3.3. The coordinates of the reconstructed phase space are $x_1(t)=s(t)$, $x_2(t)=s(t-1)$ and $x_3(t)=s(t-2)$. Such a reconstruction results in a sequence of state vectors *{y(n)}*, where $n$ plays the role of the time index, and in this case $y(1)$=[2,1,3], $y(2)$=[1,2,1], etc. Let us denote the nearest neighbor to the vector $y(1)$ as $y^k(1)$ (the superscript k indicates the kth closest neighbour),

which in this case is state vector (point in phase space) $y1(1)=y(5)$ =[2,3,1]. The distance between $y(1)$ and $y^1(1)$ is denoted by $z^1(1)$. If there exists some underlying mapping which takes $y(1)$ and $y^1(1)$, and moves them to the next time step, i.e. to the points $y(1+1)$ and $y(1,1+1)$. It is important to point out that $y(1,1+1)\neq y^1(1+1)$, which means that the closest neighbour to $y(1+1)$ is not necessarily the point $y(1,1+l)$. Similarly, the distance between $y(1+1)$ and $y(1,1+l)$ can be denoted as $z(1,1+1)$.

Table 3.3. A simple hypothetical example of phase space reconstruction used to define the nearest neighbour in order to illustrate the methodology behind estimating Lyapunov exponents from time series (adopted from Tsonis, 1992).

| $m$=3, $\tau$=1 | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| | | | | $z(1, 1+1)$ | | | | | | | | |
| | | ← | | | | → | | | | | | |
| $x_1(t)$ | 2 | 1 | 1 | 3 | 2 | 3 | 3 | 5 | 3 | 3 | 2 | 1 |
| $x_2(t)$ | 1 | 2 | 1 | 1 | 3 | 2 | 3 | 3 | 5 | 3 | 3 | 2 |
| $x_3(t)$ | 3 | 1 | 2 | 1 | 1 | 3 | 2 | 3 | 3 | 5 | 3 | 3 |
| | ↓ | ↓ | | | ↓ | ↓ | | | | | | |
| | $y(1)$ | $y(1+1)$ | | | $y^1(1)$ | $y(1,1+1)\neq y1(1+1)$ | | | | | | |
| | ← | | | | → | | | | | | | |
| | | | $z^1(1)$ | | | | | | | | | |

If $F$ is the underlying mapping, one could write:

$$z(1,1+1)=F(y^1(1))-F(y(1))=F(y(1)+z^1(1))-F(y(1)). \tag{3.65}$$

Equation (3.65) can be generalised for any point $y$(n) and any of its $k$=1,…,$l$ closest neighbours,

$$z(k, n+1)=F(y(n)+z^k(n))-F(y(n)). \tag{3.66}$$

Taylor's series expansion about $z^k(n)$ and truncation of the high-order terms (except linear) results in

$$z(k, n+\backslash)=Azk(n) \tag{3.67}$$

where the matrix $A$ is defined as

$$A=\begin{pmatrix}\frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \frac{\partial F_1}{\partial x_3}\\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \frac{\partial F_2}{\partial x_3}\\ \frac{\partial F_3}{\partial x_1} & \frac{\partial F_3}{\partial x_2} & \frac{\partial F_3}{\partial x_3}\end{pmatrix} \tag{3.68}$$

which is in fact the Jacobian matrix (3×3 in this case). Since $z(k, n+1)$ and $z^k(n)$ are vectors, one can write the first component of $z(k, n+1)$ as

$$z_1(k,n+1)=\frac{\partial F_1}{\partial x_1}z_1^k(n)+\frac{\partial F_1}{\partial x_2}z_2^k(n)+\frac{\partial F_1}{\partial x_3}z_3^k(n) \tag{3.69}$$

or considering all neighbours $k=1,\ldots,l$,

$$\begin{bmatrix}z_1(1,n+1)\\ z_1(2,n+1)\\ \vdots\\ z_1(l,n+1)\end{bmatrix}=\begin{bmatrix}z_1^1(n) & z_2^1(n) & z_3^1(n)\\ z_1^2(n) & z_2^2(n) & z_3^2(n)\\ \vdots & \vdots & \vdots\\ z_1^l(n) & z_2^l(n) & z_3^l(n)\end{bmatrix}\begin{bmatrix}\frac{\partial F_1}{\partial x_1}\\ \frac{\partial F_1}{\partial x_2}\\ \frac{\partial F_1}{\partial x_3}\end{bmatrix} \quad \text{or} \quad D=BC \tag{3.70}$$

The entries of the matrix $B$ can be estimated from the time series of the observable $\{s(n)\}$. In our case (see Table 3.3), $z_1^1(1)=s(5)-s(1)$, $z_2^1(1)=s(6)-s(2)$, etc. In general notation, $z^k(n)=\{s(n_k+(a-1)\tau)-s(n+(a-1)\tau)\}$, where $n_k$ is the time index value associated with the $k$th neighbour to $y$(n) and $a=1,2\ldots,(m-1)$, where $m$ is the embedding dimension. Likewise, the entries of the matrix $D$ can be written in general notation by the following expression $z_a(k, n+1)=\{s(n_k+1+(a-1)\tau)-s(n+1+(a-1)\tau)\}$. Therefore by inverting the matrix, in theory it is possible to obtain the partial derivatives of the mapping (entries in matrix $C$). By repeating the procedure for the second, third,…, and $n$ components all entries in equation (3.58) can be obtained. However, in practice one has more equations (due to large number of neighbours) than unknowns and thus the problem is overdeterminated. In such cases, the solution can be obtain by least-squares methods (see, for example, Eckman et al., 1986). Note that this procedure can be generalised for any embedding dimension $m$ (there will be $m$ components of the vectors). Up to this point of discussion, the presented procedure refers to just one point and its neighbours on the trajectory, and the monitoring of the evolution just one time step ahead. The complete estimation for this point requires the estimation of the $m\times m$ Jacobians for some $n$ time steps along the trajectory, which results in $A(1)$, $A(2)$,…, $A(n)$. According to the Oseledec multiplication ergodic theorem (see Abrabanel and Kennel, 1991), the Lyapunov

exponents (for the point and its neighbours) are the logarithms of the eigenvalues of the matrix

$$\lim_{n\to 0}\left\{\left[A^n\right]^{\mathrm{T}}\left[A^n\right]\right\}^{1/2n} \tag{3.71}$$

where$^{\mathrm{T}}$ denotes transpose operator and $A^n=A(1)A(2)\ldots A(n)$. Finally, the whole procedure is repeated for many other points and their neighbours. Thus, the Lyapunov exponent spectrum is produced by the average from all the points sampled along the trajectory. As pointed out by Abrabanel et al., (1989, 1990) these approaches may not provide reliable estimations for all but the leading Lyapunov exponents. The difficulty in estimating negative Lyapunov exponents is due to the fact that fractal attractors are often thin in many locations (points) along the directions of convergence of the orbits in phase space. The existence of such thin regions result in a lack of neighbours or false neighbours due to the presence of noise that can easily distort these regions. The solution of this problem is to consider quite long time series and as many as possible of the different number of points in phase space. On the other hand, if one considers neighbours that are large compared to the thickness of the attractor, yet small compared to the size of the attractor, then the points in these neighbourhoods, in general, lie close to some curved subsurface within the local neighbourhood. Employing linear mapping of the fluctuations in this case may result in a severe underestimation of the Lyapunov exponents. To overcome these problems Brown and Abrabanel (1991) showed that local nonlinear mappings (polynomials of order 3) may be advantageous in some real-life applications when one deals with noisy and limited data sets.

The procedure of estimating the Lyapunov exponents from a time series of observables presented above suggests that this task is far from trivial and computationally demanding. The monitoring of the long-term evolution of the fluctuations from nearby orbits on the trajectory implicitly requires modelling of the dynamics of the attractor (that is the mapping from one state to another), though in the stage of identifying the underlying dynamics of the system. Thus, estimation of the complete spectrum of the Lyapunov exponents should be seen as an iterative and interactive process, which sometimes requires from the modeller a re-estimation of the complete geometrical properties of the attractor in the modelling process, until there is enough convincing evidence for their reliable estimation. In this work we have used the modified version of the algorithm of Brown and Abrabanel (1991) for Lyapunov spectrum estimation. Furthermore, the largest Lyapunov exponent, which has significant dynamics identification and modelling implications, was checked against the algorithm of Wolf et al. (1985). As an example, the Lyapunov spectrum estimated from the water level time series at Hoek van Holland tidal station is presented in Figure 2.28.

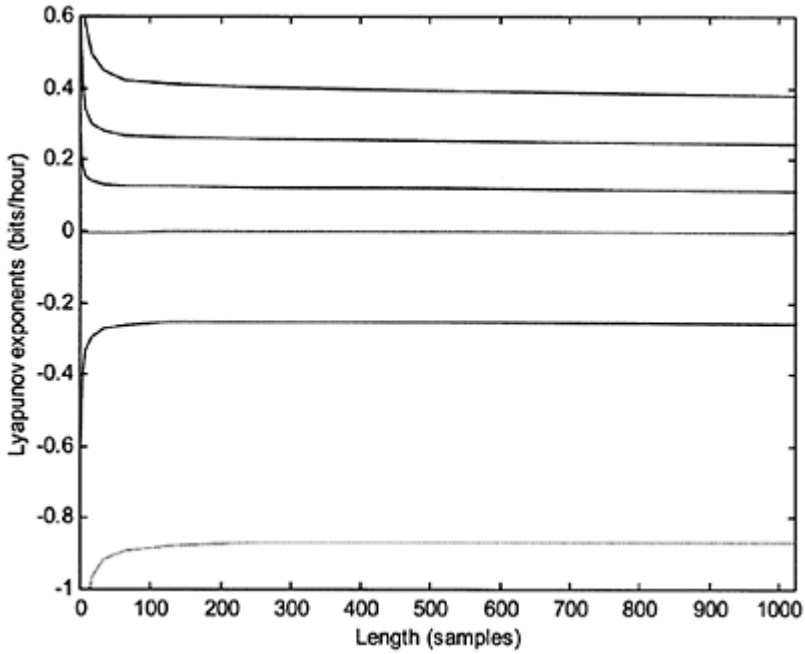



Figure 3.28. Estimated average local Lyapunov exponents for the hourly water level time series at Hoek van Holland tidal station in $m$=6 dimensions. The data are consistent is showing a sum of global Lyapunov exponents (the values for about 1000 steps along the attractor) that is negative.

The largest Lyapunov exponent is estimated as $\lambda_1$=0.38 (uncertainty 0.02) which indicates a loss of information of 0.38 bits/hour during the dynamical evolution of the system, and thus loss of predictive capabilities. The reliable limits of predictability of the system based on the available time series is between $\lambda_1^{-1}$=1/0.38=2.63 hours and $\tau/\lambda_1$=4/0.38=10.53 hours. The Lyapunov spectrum contains a large negative exponent $\lambda_6$=−0.90 which indicates presence of strong dissipation mechanisms in the dynamics of the system. The presence of positive Lyapunov exponents and the fact that $\sum_{n=1}^{6}\lambda_n = -0.40 \; < 0$, provide strong evidence that dynamics of the system is driven by deterministic chaos. Furthermore, one of the Lyaunov exponents $\lambda_4$=0.0 is clearly zero, which indicates that the deterministic motion of system can be mathematically

described by a system of 6 nonlinear ordinary differential equations. More results are presented in Chapter 6.

### *3.3.5 Entropies form time series data*

When we introduced the concept of mutual information for time delay estimation, we have also briefly discussed the concept of information entropy. The general concept of entropy is fundamental for the study of statistical mechanics and thermodynamics. From the thermodynamics perspective entropy is a quantity describing the amount of disorder in the system. One can generalise this concept to the amount of information stored in more general probability distributions. This is the entropy approach that *information theory* is concerned with. The theory has been developed since 1940s and 1950s, where the main contributors were Shannon, Renyi and Kolmogorov. The text book of Jaynes (1995) gives an historical overview of the development of the information theory and its connections with the probability theory complemented by an excellent philosophical discussion.

Information theory provides an important approach to time series analysis. The observation of a system is regarded as a source of information, that is, a stream of numbers which can be considered as a transmitted message. If these numbers are distributed according to some probability distribution, and the transitions between different numbers occur with well-defined probabilities, one can questions of type: "How much do I learn on average about the sate of the dynamical system when I perform exactly one measurement?", or "How much information do I know about the future observations when I have observed the entire past?". Information theory supplies concepts that can give certain quantitative answers. For example, when one knows that the dynamical system is at rest at a stable fixed point, a single observation suffices to determine the whole future with exactly the same precision as the precision of the past observations. If one deals with regular periodic dynamical systems, an observation of a single period is enough to know all about the time series generated by this system. If the system is random, then one is not able to predict with certainty the next observation even with an infinite number of previous observations. One could also ask whether the concept of entropy could provide certain quantitative information when one deals with systems whose dynamics is quasi-periodic and chaotic. Therefore, for these reasons a numerical value of the entropy of a time series observed on a certain dynamical system is interesting for its characterisation. Firstly, its inverse (similar to Lyapunov exponents) is the relevant time scale or the predictability of the system.

Secondly, it suplies topological information about the folding process of the attractor. Thirdly, in general, it can provide qualitative information for identification of the dynamics of the system, i.e. it is zero for a systems which exhibit regular periodic motion, positive and finite for deterministic chaos and infinite for stochastic processes.

One of the most commonly used entropy estimation in nonlinear time series analysis is the Kolmogorov-Sinai entropy $h_{KS}$, which can be obtained from the set of correlation functions $C_m(r)$ (Grassberger and Procaccia, 1983). For practical applications, it can be approximated as the limit as the embedding dimension $m \to \infty$ of the distance (in log-log coordinates) between successive correlation curves $C_m(r)$ and $C_{m+1}(r)$ (see Baddi and Politi, 1985 and Grassberger, 1985):

$$h_2(m) \cong \lim_{r\to 0}\left(\frac{1}{\Delta t}\left[\log C_m(r) - \log C_{m+1}(r)\right]\right) \tag{3.72}$$

and further

$$h_2 \cong \lim_{m\to\infty}\left[h_2(m)\right] . \tag{3.73}$$

Similary to the generalisation of the dimensions of the attractor (see Section 3.2.4), the entropy concept can also be generalised. We have discussed that the most robust dimension is the correlation dimension $D_2$, and the same goes for the entropies: $h_2$ is the most robust, due to the fact that the second moment in the correlation sum is an arithmetic average over the numbers of the neighbours. In this case, $h_2$ is just an estimate for $h_{KS}$, and for multifractal attractors can be considerably smaller than $h_{KS}$ (reported by Grassberger and Procaccia, 1983).

One of the main difficulties of extracting the entropies from time series data is primarily because their computation requires far more data than calculating dimensions and Lyapunov exponents, since the limit $m\to\infty$ constitutes the crucial problem (high embeddings are needed). However, as discussed above $h_2$ can be approximated using the correlation sum, which is anyway computed for the estimation of the correlation dimension. Also, we recall that using the Pesins' identity $h_{KS} = \sum_{i:\lambda_i>0}\lambda_i$, one can find the upper bound of the Kolmogorov-Sinai entropy.

### *3.3.6 Nonlinear noise reduction*

All data to some extent are contaminated by noise, and noise by definition is the unwanted part of the data. The only question is what does it mean for practical analysis tasks such as modelling and prediction. Generally speaking, in the terms of a nonlinear time series analysis the effect of noise is one of the most prominent limiting factors for the predictability of deterministic systems. The range of the length scale through which one can monitor the exponential divergence of nearby trajectories is bounded from below by noise in the data. An extreme consequence of the noise is the breakdown of the self-similarity and the fractal nature of the attractor of a deterministic dynamical system that could lead to a wrong impression about the qualitative behaviour of the dynamics. We have already discussed the effect of noise in the correlation dimension estimation, especially in terms of the difficulties of obtaining the correlation exponent at microscopic (small) length scales. Several authors have studied the effect of noise on the estimation of the geometrical and dynamical invariants estimation within the nonlinear time series analysis context (see, for example, Schreiber and Kantz (1995). They presented a remarkable result, concluding that in case of presence of uncorrected white noise the tolerable noise level for estimation of various dimensions, exponents and entropies from time series, focusing on the small length scales, cannot be greater than 5%. Therefore one has to deal with the sensitivity of these quantities to noise, especially focusing on the length scales which give a robust estimation of those quantities. Another approach is to try to reduce the noise level using appropriate noise reduction algorithms. Before

proceeding with the discussion on noise reduction, we have to make an important distinction between the terms used. In general, there are two types of noise that one can clearly define: measurement and dynamical noise. *Measurement noise* refers to the corruption of observations by errors which are independent of the dynamics. The dynamics satisfy some deterministic mapping $\mathbf{x}_{n+1}=F(\mathbf{x}_n)$, but one measures the scalars $s_n=s(\mathbf{x}_n)+\eta n$, where $s(\mathbf{x}_n)$ is some smooth measuring functions which maps points on the attractor to real numbers and $\eta_n$ are random numbers. The series $\{\eta_n\}$ is referred as to the measurement noise. On the other hand, *dynamical noise* is a feedback process wherein the system is perturbed by a small random amount at each time step, i.e.

$$\mathbf{x}_{n+1}=F(\mathbf{x}_n+\eta_n). \tag{3.74}$$

Dynamical and measurement noise are two notions that are very difficult to distinguish *a posteriori* based on the data only, and furthermore they can be mapped onto each other, as has been pointed out by Bowen and Ruelle (1975). Although one is interested in quantifying the dynamical noise, generally dynamical noise induces greater problems in the nonlinear time series analysis than the measurement (or additive) noise, since it can pollute and distort a nearby clean trajectory of the underlying deterministic system. Furthermore, what one interprets as a dynamical noise may sometimes be higher-dimensional deterministic parts of the dynamics with small amplitudes. Therefore, dynamical noise may have great influence on the observed dynamics because transitions to qualitatively different behaviour (bifurcations) can be induced or delayed by the dynamical noise.

Noise reduction, in a practical sense, means that one tries to decompose a time series into two components, one of which evidently contains the deterministic signal and the other contains the random fluctuations. The classical linear statistical tool to carry out this decomposition is the power spectrum. Random noise has a flat and broadband spectrum, whereas periodic and quasi-periodic time series exhibit sharp spectral lines. After both components have been identified in the time series, one could use any linear filter to separate the time series accordingly. However, this approach fails for deterministic chaotic dynamical systems because the output of such systems usually leads to a broadband power spectra itself and thus possesses spectral properties generally attributed to random noise. Even if parts of the spectrum can be clearly associated with the deterministic nature of the signal, a separation of the noise for most parts of the frequencies will fail.

The way to exploit the deterministic structure using nonlinear noise reduction techniques is closely related to the modelling and prediction of the attractor of the dynamical system in the reconstructed phase space from time series. The main idea behind this is the following: Let the time evolution of the dynamical system be deterministic with the mapping $x_n=f(x_{n-m},\ldots, x_{n-1})$, however, not known to us. All the information we have about the system is a time series of some noisy measurements observed on the dynamical system $s_n=x_n+\eta_n$, where $\eta_n$ is suppose to be random noise characterised by fast decay of the autocorrelation function and no correlations with the signal $x_n$. If the time series is free of noise, than the trajectory of the deterministic system in the phase space will define a clear geometrical object. However, due to the presence of noise the points in the reconstructed phase space will not lie on the “true” trajectory, but

will be scattered around it. In order to clean a particular value of the time series, one has to replace (correct) the measurement by a prediction $\hat{x}_n$, based on the previous measurements in the reconstructed phase space.

$$\hat{x}_n = \hat{f}(s_{n-m},\ldots,s_{n-1}) \tag{3.75}$$

This idea can be further enhanced using an implicit relation (Kantz et al., 1993), such as

$$x_n - f(x_{n-m},\ldots, x_{n-1})=0 \tag{3.75}$$

and solving it for one of the coordinates in the middle, say $x_{n-m/2}$. Figure 3.29 gives a graphical representation of the simple nonlinear noise reduction technique.

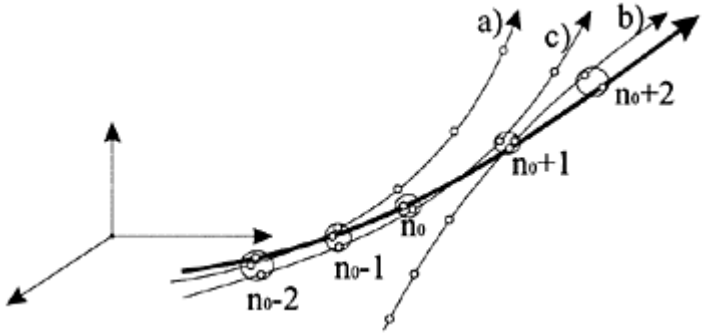



Figure 3.29. Graphical illustration of the simple nonlinear noise reduction technique. With regard to the reference time $n_0$, the trajectory is close to: a) the true trajectory (bold line) for two time steps in the past; b) the true trajectory for two time steps in the future. Both are not close with regard to the time $n_0$, due to the sensitive dependence on the initial conditions. Finally, the trajectory c) is close in the past and in the future to the true trajectory and thus to the time $n_0$. (Figure courtesy of Kantz et al., 1993).

Of course, one does not know the mapping function *f* and one must approximate it. Based on the type of the approximation of the mapping function *f*, several versions of this nonlinear noise reduction approach exist. The basic approximation can be that the

mapping function *f* is locally constant. In other words, in order to obtain the estimate $\hat{x}_{n_0-m/2}$ for the value $x_{n_0-m/2}$, one could form delay vectors (points in phase space) $\mathbf{y}_n=(s_{n-m+1},\ldots, s_n)$ and find which ones are close to the $\mathbf{y}_{n_0}$. Then the average value of the $s_{n-m/2}$ is used as a cleaned (corrected) value $\hat{x}_{n_0-m/2}$:

$$\hat{x}_{n_0-m/2} = \frac{1}{\left|U_\varepsilon(\mathbf{y}_{n_0})\right|} \sum_{q \in U_\varepsilon(\mathbf{y}_{n_0})} s_{n-m/2} \tag{3.76}$$

where $U_\varepsilon(\mathbf{y}_{n_0})$ the neighbourhood of radius ε around the point $\mathbf{y}_{n_0}$. This formula is very similar to the local modelling and prediction of the dynamics of the attractor (see next Section 3.3.7), with the main difference that here one can use the future values of the time series, which practically means that the neighbourhood is never empty. The only parameter is used in (3.76) is the radius ε and if ε is too small, in the worst case, what can happen is that no correction of the observation is made at all. If the radius ε is too large, the outcome of the algorithm can result in a slight distortion of the geometry of the attractor. In practical applications one must ensure that the distortion is considerably smaller than the noise level in the time series. Obviously, before applying any noise reduction or surgical noise removal from the time series in real applications, one should try these algorithms on well-known mathematical dynamical systems. As an illustration, we present here example of the application of simple nonlinear noise reduction algorithm on the Hénon map. The “clean” signal obtained by 15000 iterations of the Hénon map was polluted by 10% of additive, independent and uniformly distributed noise, which was bounded in magnitude (0.10) with standard deviation of 0.07 (compared to the standard deviation of the generated time series of 0.72). Figure 3.30 shows the reconstructed phase space of the polluted and cleaned time series.

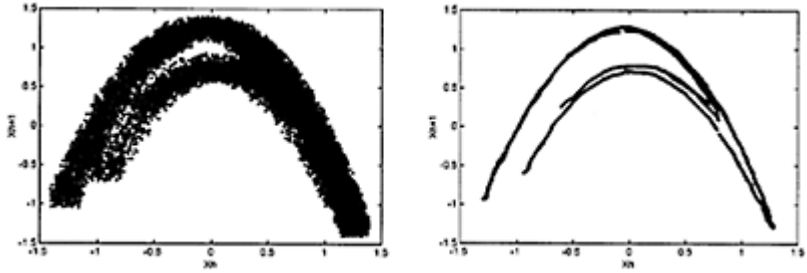


Figure 3.30. Hénon map time series (15000 points) of polluted by 10% additive noise and the cleaned time series by applying the simple nonlinear noise reduction algorithm.

As an example of a real application, Figure 3.31 shows the correlation exponent estimation as a function of the embedding dimension for daily rainfall time series (period 1955–1998, 16071 samples) at De Bilt meteo station in the Netherlands, for both the original and "cleaned" time series.

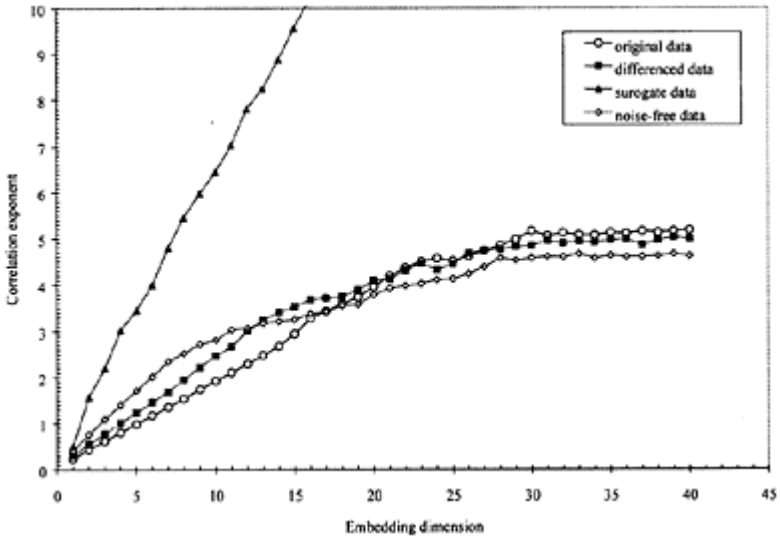



Figure 3.31. Relation between the correlation exponent and the embedding dimension for the daily rainfall time series data at De Bilt meteo station (period 1955–1998). Time delay $\tau$=4 hours.

From the results presented on Figure 3.31 one can see that the "noise-free" time series of the daily rainfall exhibits better saturation with the increase of the embedding dimension and lower correlation dimension (4.65 in this case) compared to the original time series. Correlation dimension estimation was also performed on the derivative time series of the rainfall data (daily intensities) and on stochastic surrogate time series containing the same power spectrum as the original time series. As expected, the stochastic surrogate data does not exhibit any saturation with the increase of the embedding dimension. This indicates that the underlying dynamics of the daily rainfall times series at De Built meteo station may be driven by deterministic chaos. More discussions and results are presented in Chapter 6.

### *3.3.7 Modelling and forecasting (global vs. local)*

The most direct link between the mathematical methods and techniques offered by the theory of nonlinear dynamical systems and the concept of deterministic chaos and the real world dynamical systems (such as aquatic systems) is the analysis of data (time series) produced by those systems. Having introduced and demonstrated the methods and techniques for reconstruction, identification, delineation and quantification of the underlying dynamics of such nonlinear systems from observables, one may justifiably get the feeling that subjective judgement of the modeller may be required in order to make maximal use of those techniques; thus one requires interactive analysis process. Once the dynamics of the system is reconstructed and characterised from the time series of observables, the next natural step is to explore possibilities for constructing models from the data that will realistically model the underlying attractor dictating the dynamics of the system. In general, the ultimate goal of constructing such models is forecasting, which in the terms of the phase space representation of the dynamics means the extrapolation of the trajectory, thus, modelling the dynamical evolution of the system in time which is yet to be observed. Therefore, in this context, the concept of learning models from data is usually a nonlinear regression estimation of the reconstructed trajectory of the dynamical system from time series data in phase space. These regression estimation techniques have already been discussed in Chapter 1. We have also learned that the dissipative dynamical systems, though deterministic in nature, exhibit sensitive dependence on initial conditions and exponential growth of small dynamical disturbances in time, and therefore can be characterised by deterministic chaos, which in turn limits the predictability. One could certainly pose the following questions: What are the consequences of the existence of such chaotic dynamics? or How can such dynamical systems best be modelled? or How uncertain is the forecasting?

In order to give answers to these (and similar) questions, one should not forget that the chaotic dynamics is deterministic. Chaotic dynamical systems obey certain rules. Their dynamics live in a certain phase space with certain degrees of freedom, and are asymptotically stable and attracted to certain geometrical objects that can be identified and quantified using various geometrical invariants and measures, such as the dimensions. They have limited predicting power, which clearly and explicitly can be quantified with the Lyapunov exponents and entropies, for example. However, before their predictive power is lost (i.e. for short-term prediction) their predictability may be adequate and even better than the generally applied nonlinear neural network prediction. This advantage is due to the knowledge gained by the reconstruction of the underlying determinism, and the capability of accurately modelling the evolution of the nearby orbits in the reconstructed phase space locally, rather then globally. Therefore, the modelling in this sense is the modelling of the reconstructed phase space of the dynamical system. Consequently, the basic philosophy behind nonlinear forecasting is the same as that of estimating the Lyapunov exponents (described in Section 3.3.4): to accurately obtain from a time series of an observable the mapping that dictates where in an $m$-dimensional phase space the next point (state) will be located.

To model nonlinear deterministic dynamics, or a dominant deterministic part of some mixed system, one has to accurately reconstruct the phase space from time series of observable. At this point of the discussion, we consider an $m$-dimensional delay embedding based on univariate (scalar) time series of an observable; we extend this latter

to vector valued (multivariate) time series embedding. Since the time series data are discretely sampled over time, the underlying dynamics is described by a deterministic model in phase space, which is always a map of the form

$$\boldsymbol{Y}_{n+1}=f_n(\boldsymbol{Y}_n) \tag{3.77}$$

where $\boldsymbol{Y}_n$ are delayed vectors (states) $\boldsymbol{Y}_t=\{s_n, s_{n-\tau}, s_{n-2\tau},\ldots, s_{n-(m-1)\tau}\}$, formed by the embedding of the time series of observable $\{s_n=x_n+\eta_n\}$ in $m$-dimensional phase space with an appropriate time delay $\tau=v\Delta t$ (v is time index—integer). In order to forecast the next state of the dynamical system, one needs find the estimator of the regression function $\hat{f}$, and thus, one can estimate the prediction of $s_{n+1}$,

$$\hat{s}_{n+1} = \hat{f}_n(\boldsymbol{Y}_n) \tag{3.77}$$

After these more general considerations, the next step is to find the proper approximation of the model, expressed through its structure and capacity, and a criterion for the quality of the model which is to be learned form the data in the reconstructed phase space (such as the ERM or SRM principles discussed in Chapter 1). Generally speaking, there are two possibilities for choosing the structure of the model in order to approximate the "true" mapping function, namely *global* and *local* approximations.

## GLOBAL MODELS IN PHASE SPACE

The global modelling in phase space is a global nonlinear regression estimation problem, which we addressed in Section 2.6. We basically have to choose appropriate parametric or nonparametric model (functional form), which has enough capacity to approximate the true (unknown) function of the whole attractor. A widely used approach is to choose the structure of the function f to be a superposition of several basis functions, $f = \sum_{i=1}^{k} \alpha_i \Phi_i$. The $k$ basis functions $\Phi_i$ usually are kept fix during the empirical or structural risk minimisation procedure, while optimising the parameters $\alpha_i$ of the model. The most commonly used basis functions range from polynomials, radial basis functions, sigmoid functions (neural networks) and recently wavelet basis functions.

The main advantages of *polynomials* are that most practitioners are familiar with them, there is the possibility of estimating the parameters using linear algebra and the existence of physical interpretations of the trained model. However, the main drawback is that the number of parameters may become very large, for example, in $m$-dimensional phase space if one uses polynomial of order $l$, the number of parameters is $k=(m+1)!/m!l!$. As an illustrative step-by-step example of estimating polynomial mapping in phase space in order to model the attractor and forecast the future value of the observable, we consider a very simple setup: We are given a time series of observable $s(n)$ (which were in fact generated by the logistic map) of 18 observations in total, $s(1)=0.4100$, $s(2)=0.9676,\ldots$, $s(18)=0.9979$. Using the time delay embedding method (with $\tau=1$ and $m=2$) one can reconstruct two dimensional phase space from the time series with coordinates $y_1(n)=s(n-\tau)$ and $y_2(n)=s(n)$. Note that the optimal time delay and the embedding

dimension are estimated using the methods and techniques for phase space reconstruction we described earlier in this chapter. Such reconstruction results in a sequence of state vectors (points) y(n) in phase space that lie on the trajectory that has to be modelled; see Figure 3.32. For example, point 2 is defined by the state vector $y(2)$ with coordinates (0.41, 0.9676), point 3 by $y(3)$=(0.9676, 0.1254), etc., and the last point is point 18 defined by the vector y(18)=(0.4775, 0.9979). The main task now is to obtain a prediction for the observation s(19), i.e. one has to find the next point in the phase space $y(n+1)=f(y(n))$. From the reconstructed phase space it is visible that the trajectory of dynamical system can be approximated by a quadratic polynomial function, e.g. $y(n+1)=a+by\ (n)+cy^2\ (n)$.

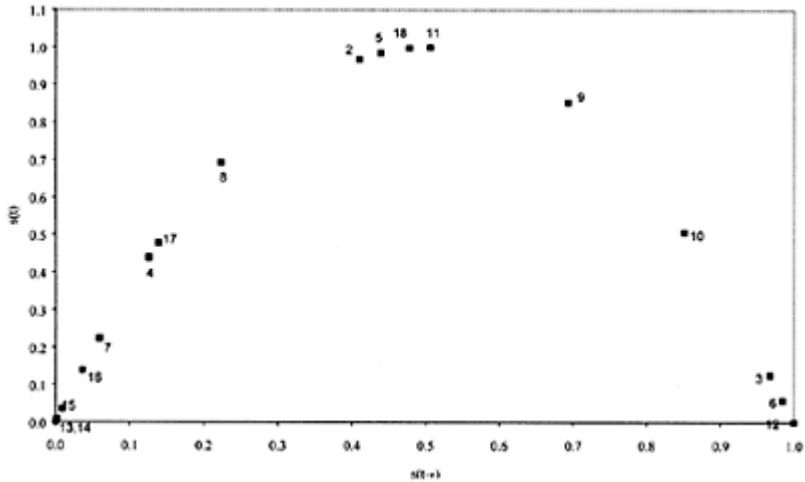


Figure 3.32. Illustration of two-dimensional reconstruction of the phase space from a time series of observable and global modelling in phase space on a hypothetical example.

The task now is to estimate the parameters $a$, $b$ and $c$ of the global mapping function in phase space. However, since there are more points which must satisfy the mapping function than parameters, the problem is overdetermined. In this example, one could write 16 equations mapping the states from one time step to another and 3 parameters, namely

$y(3)=a+by(2)+cy^2(2)$
$y(4)=a+by(3)+cy^2(3)$
…
$y(18)=a+by(17)+cy^2(17)$

which can be written in a matrix form as $A=BC$, where the entries in matrix $C$ are the parameters of the function. Due to overestimation one cannot directly invert the matrix $B$,

i.e. $C=AB^{-1}$. In this case one could employ any nonlinear regression estimation technique discussed in Chapter 2. Strang (1986) showed that by using the squared loss (error) function $e=\|BC-A\|^2$ and minimising the empirical risk, the parameter vector can be estimated as $C=(B^{\mathrm{T}} B)^{-1} B^{\mathrm{T}}A$. In this example, parameters are estimated as $a\approx 0$, $b\approx 4$ and $c\approx -4$. Thus, the global mapping function which can be used to forecast the next states of the dynamical system is of the form

$$y(n+1)=4y(n)-4y^2(n)=4y(n)[1-y(n)]$$

which is in fact the logistic map with parameter $\mu=4$.

Another very flexible model class for approximating the global mapping function in phase space, which was introduced in the field of nonlinear time series analysis by Broomhead and Lowe (1989), is the *radial basis functions*. The basis function in this case is a scalar function $\Phi(r)$ of the argument $r$. One has to further select $k$ centres $\mathbf{y}_i$ on the reconstructed attractor. In this case the mapping function can be written as

$$f(\mathbf{x}) = \alpha_0 + \sum_{i=1}^{k} \alpha_i \Phi\left(\left\|\mathbf{x}-\mathbf{y}_i\right\|\right) \tag{3.78}$$

Typical basis functions $\Phi(r)$ that are most commonly used are bell-shaped, though increasing and even singular functions have been reported in the literature. The function $f$ is modelled by adjusting the parameters $\alpha_i$ of the basis functions. If the centres $\mathbf{y}_i$ are reasonably well distributed on the attractor, the superposition of the basis functions yields a global hyper-surface which models the global dynamics of the attractor. Thus, the number and the positions of the centres have to be selected properly. In order to assist this procedure, one can use some clustering algorithm (such as *k*-means) to initialise the centres. Determining the parameters $\alpha_i$ is then a risk minimisation problem. The typical width of the basis functions $\Phi(r)$ can be optimised systematically by testing several values. Furthermore, the number of the centres and the width of the basis functions can be introduced as regularisation parameters in the optimisation algorithm, for example, gradient descent.

*Neural networks* provide another nonparametric class of models for the global modelling in phase space and have already been discussed in Section 2.6.3.

The last class of models that we would like to address in the context of global modelling in phase space is the so-called *wavelet networks*. The wavelet network can be seen as a special feed forward neural network supported by the continuous wavelet theory. The have been recently introduced by Zhang (1993) and started to attract much attention due to their capabilities of efficiently decomposing the time series. The basic idea of the wavelet transform is to map functions from the amplitude-time domain to a frequency-time phase space. For the sake of clarity we first discuss a one-dimensional case in $\boldsymbol{R}$ and then its generalisation to $\boldsymbol{R}^n$. The wavelet transform is the transformation (or projection) of the original function to a family of functions generated by dilating and translating a single basis function which is called a *mother* (or basic) *wavelet*. In mathematical terms, if $\psi(x)$ is the mother wavelet, the family

$$\left\{ d^{\frac{1}{2}} \psi(d(x-t)) : t \in \boldsymbol{R}, d \in \boldsymbol{R}_+ \right\} \tag{3.79}$$

is used for the *continuous wavelet transform*. If the mother wavelet is chosen such that

$$C_\psi \cong \int_R \frac{|\hat{\psi}(w)|^2}{|w|} dw < \infty \tag{3.80}$$

where $\hat{\psi}(w)$ is the Fourier transform of $\psi(x)$, then the following transformations hold for any $f \in \boldsymbol{R}$:

$$W(d,t) = \int_R f(x) d^{1/2} \psi\big(d(x-t)\big) dx \tag{3.81}$$

and

$$f(x) = \frac{1}{C_\psi} \int_R \int_{R_+} W(d,t)\, d^{1/2} \psi\big(d(x-t)\big)\, \mathrm{d}d\, \mathrm{d}t \ . \tag{3.82}$$

The expression (3.80) is called the continuous wavelet transform and by using the convolution integral it calculates the set of wavelet coefficients for values of the different scaling (dilating) and translating parameters. The inverse of the scaling parameter defines the presence of particular frequencies, and the translating parameter incorporates the time structure in the transformation. The expression (3.81) is the *reconstructed function* on the basis of the wavelet transformation. Note that this reconstruction procedure can be utilised to extract or filter out particular frequencies from the signal (this is practically illustrated in the application in the Section 6.2). Furthermore, the choice of the mother wavelet $\psi(x)$ determines the properties of the wavelet transform (see Daubechies, 1990 for discussion). In order to implement practically the continuous wavelet transform and the reconstruction respectively, they have to be written in discretised form. Normally in the wavelet theory literature (Lewalle, 1995) this is done by taking a regular lattice with a uniformly distributed translation parameter $t$ and exponentially distributed scaling dilation parameter $d$, or more precisely, the following discretised wavelet family is used

$$\Psi \cong \left\{ \psi_{s,n} = \alpha^{-s/2} \psi\big(\alpha^{-s}(x - \beta n)\big) : s, n \in \boldsymbol{Z} \right\} \tag{3.83}$$

where $\alpha$ and $\beta$ define the steps of the scale $(s)$ and the translation $(n)$ discretisations respectively. As pointed out by Daubechies (1990) by using this discretised wavelet family, the reconstruction expression, an analogue of (3.82), will not automatically hold. An additional condition for the wavelet family (3.38) is the need for it to fold, so that the discretised wavelet family constitutes a *frame* of the $\boldsymbol{R}$ space (see details in Daubechies, 1990). If the wavelet family (3.83) is a frame of the $\boldsymbol{R}$ space, then an analogue of the reconstruction expression (3.82) can be written as:

$$f(x) = \sum_{s,n} W(s,n)\, \alpha^{-s/2} \psi\big(\alpha^{-s}(x - \beta n)\big) \tag{3.84}$$

The expression (3.84) can be further written as a summation of $N$ approximations of several discretised wavelet family (3.83), or in general form:

$$f(x) \approx \sum_{i=1}^{N} w_i \psi\left(d_i(x - t_i)\right), \quad w_i, d_i, t_i \in \boldsymbol{R} \tag{3.85}$$

The expression (3.85) defines the *one-dimensional wavelet network* which can be only used to approximate functions in $\boldsymbol{R}$. In order to generate an approximate function in $\boldsymbol{R}^n$, we need to construct multivariate wavelets. Zhang (1994) presented an extension of the one-dimensional wavelet network (3.85) using radial wavelets of form:

$$\psi(x) = \phi\|x\| \tag{3.86}$$

where the norm is $\|x\|=(x^{\mathrm{T}}x)^{1/2}$ and $\phi$ is a single variable nonlinear function. Using the radial wavelet (3.86) the one-dimensional wavelet network can be extended to the following multi-dimensional function approximator:

$$g(x) = \sum_{i=1}^{N} w_i \psi\left(\mathrm{diag}(d_i)(x - t_i)\right) \tag{3.87}$$

where $N$ is the number of the wavelets used to construct the network, $d_i \in \boldsymbol{R}^n$ are scaling parameters (diagonal matrices), $t_i \in \boldsymbol{R}^n$ are the translation parameters and $w_i \in \boldsymbol{R}$ are linear weights of the wavelets. Similary to the neural network, the parameters of the wavelet net (the linear weights of the contribution of each of the wavelets) can be learned by minimising the loss function using any optimisation algorithm, such as described in section 2.9.

## LOCAL MODELS IN PHASE SPACE

In general, global models provide good approximations of the mapping function if $f$ is well behaved and not very complicated. For dynamical systems which exhibit chaotic determinism whereby close orbits in the phase space diverge exponentially locally, a better approach is building *local models* in phase space, as introduced by Farmer and Sidorowich (1987) and then further elaborated by Casdagli (1989) and Sugihara and May (1990). The basic idea of the local approximation methods is to use only the states close to present state in phase space in order to make predictions. Thus, they learn neighbourhood relations from the data and map them forward in time. Although this approach is conceptually simpler than the global models, depending on the type of the local approximations used, they can require a large computational effort. In general, local approximations are well suited for long time series (the phase space is well populated with neighbours) and small noise level.

In order to predict the value of the observable $S_{n+T}$ ($s_n \approx x_n$ for low noise level), which is part of the state vector $Y_{n+T}$ where $T$ is some time horizon in the future, based on the state vectors $\boldsymbol{Y}_n$ and past history embedded in the reconstructed phase space, $k$ nearest

neighbours of $\boldsymbol{Y}_n$ are found on the basis of some norm $\|\boldsymbol{Y}_n-\boldsymbol{Y}_{n'}\|$, with $n'<n$. Depending on the number of the neighbours considered and the type of the local mapping chosen, several variations of the local approximation method are attempted:

(i) Zeroth order approximation in which the closest neighbour to the current state in phase space is chosen and the prediction is simply given as

$$\hat{Y}_{n+T}=\boldsymbol{Y}_{n'+T} \tag{3.88}$$

where $n'$ is the time of the closest neighbour. In the example previously introduced (see Figure 3.32) the task was to obtain a prediction for the observation $s(19)$. According to this approximation, s(19) is approximated to the point to which the nearest neighbour in the phase space of y(18) evolved. In this example, the nearest neighbour to state y(18) is state y(11), which evolved to y(12), whose coordinates are (0.9998, 0.000568). Thus, the prediction for s(19) is simply s(19)≈0.000568. The true value of s(19), computed from the logistic map is s(19)=0.000808, which is a reasonable prediction. Having the value of s(19) and thus the state y(19), one can find its nearest neighbour and predict the value of s(20), and so on. Such zeroth order local approximation was initially applied by Lorenz, who tried to predict the weather using this kind of *persistent* prediction. In order to predict the weather tomorrow one looks in the past to find closest weather pattern to that of the current weather, further analyse how it evolved and assume that this will be the prediction of the weather for tomorrow. Normally a question arises whether one neighbour is enough for such a prediction.

An improvement to the zeroth order approximation is to consider several neighbours in phase space. To predict the future of a point in phase space, one searches for its $k$ closest neighbours and uses the average of the images of all these points, i.e.

$$\hat{Y}_{n+T}=\frac{1}{k}\sum_{i=1}^{k}\boldsymbol{Y}_{n_i'+T} \quad . \tag{3.89}$$

The number of the neighbours $k$ can be obtained by minimising the prediction error (optimisation), space time separation plots or false nearest neighbours algorithm. A modification to this local approximation is to introduce further a weighted average of the images of several neighbours in the form

$$\hat{Y}_{n+T}=\frac{\sum_{i=1}^{k}w_i\boldsymbol{Y}_{n_i'+T}}{\sum_{i=1}^{k}w_i} \quad . \tag{3.90}$$

Sugihara and May (1992) further introduced exponential weighting of images of exactly $k=m+1$ neighbours, which forms a simplex containing the current point

$$\hat{Y}_{n+T} = \frac{\sum_{i=1}^{k=m+1} \exp(w_i) \boldsymbol{Y}_{n_i+T}}{\sum_{i=1}^{k=m+1} \exp(w_i)} . \tag{3.91}$$

There are also other similar suggestions of using this kind of local approximation, which in fact exploits the idea of *almost model free* local approximations.

(ii) Local linear models (LLMs)

An improvement to the zeroth order approximation is the first order or *local linear approximation*. The idea is to consider the $k$ neighbouring points and learn local linear mapping functions in order to make predictions. The predicted state can be expressed as

$$\hat{Y}_{n+T}=f_n(\boldsymbol{Y}_n)=\mathbf{A}(\boldsymbol{Y}_n)+\mathbf{B} \tag{3.92}$$

where $m \times m$ matrix is the Jacobian of $f_n$ at $\boldsymbol{Y}_n$ and $\mathbf{B}$ is an m-vector. The solution of equation (3.92) is discussed in section 2.5.6. Following the example that we have introduced, in order to predict s(19) one can find the closest neighbours to its state, namely, the closest neighbours to the state y(18) are points 2, 5 and 11. We can further fit a linear function between the closest neighbours and their images (the states towards they evolved; 3, 6 and 12), such as

$$\begin{aligned} y(3)&=a+by(2) \\ y(6)&=a+by(5) \\ y(12)&=a+by(11) \end{aligned} \tag{3.93}$$

By solving (3.93) we estimate the parameters $a$ and $b$ and thus predict the next state, $y(19)=a+by(18)$ (therefore the value of s(19) as well). The predicted value of s(19) by local linear approximation is s(19)=0.00887, which is better than the zeroth order approximation. In the same manner, we further find the neighbours of the point 19 and reestimate the local mapping function. This type of adaptive local approximation is referred to as *direct forecasting* and requires constructing the local mapping function at each prediction time step. An alternative to this approach is to obtain the local mapping function $f_n$ and then to iterate the predictions some time steps in the future until the prediction horizon $T$ is reached ($T=i\Delta t$, $i=1,2\ldots,$). Such *iterative forecasting* is simpler and sometimes can give better results than direct forecasting, especially for longer prediction horizons.

(iii) Higher order approximations (polynomials)

Analogous to the local linear approximation, one can increase the order of the local approximations using high order polynomials and even neural networks. However, this dramatically increases the number of parameters to be estimated and reduces the robustness of the local approximation, since at each time step one has to reestimate a neural network, for example, without much gain in prediction accuracy (see Abarbanel, 1996). Some authors (see, for example, Farmer and Sidorowich, 1988) have also proposed an estimation of the error associated with iterative forecasting:

$E \approx N^{-(o+1)/m} e^{\lambda_1 T}$, where $m$ is the embedding dimension, $N$ number of point in the time series, $o$ is the order of local approximation, $\lambda_1$ is the largest Lyapunov exponent and $T$ is the prediction horizon.

As an illustration of the forecasting performances of the local linear models in phase space, in Figure 3.33 we present some of the results obtained by a nonlinear time series analysis of the hourly surge water levels at Hoek van Holland tidal station in the Netherlands.

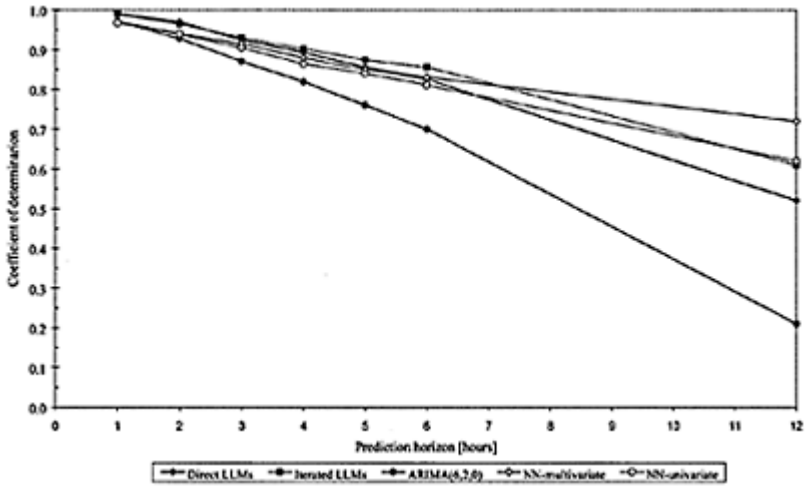



Figure 3.33. Comparison between performance of different types of prediction models for the surge water level data at Hoek van Holland tidal station. Coefficient of determination (squared correlation coefficient) between the measured and predicted data is used as a performance measure.

From the results presented in Figure 3.33, we can see that for short-term predictions (up to 6 hours) local linear models in phase space, both direct and iterated, perform better in comparison to nonlinear neural networks and linear ARIMA models. The local linear models were calculated using the reconstructed phase space (embedding dimension $m$=5 and time delay $t$=4) from the time series of hourly surge water levels (1990–1996) by utilising the knowledge of the past evolution of $k$=300 neighbouring states. Two neural networks were considered in this case. The first neural network was constructed using the same information of the reconstructed phase space, thus generating a global nonlinear model of the phase space. The second neural network was constructed using time series data of water levels, air pressure, wind speed and wind direction from several

neighboring tidal stations (see Solomatine et al. 1999). LLMs clearly show better results than both neural networks for short prediction horizons (up to 6 hours), which justifies the predictability of the dynamical system obtained from the entropies and Lyapunov exponents from the time series. Finally, as a comparison between the nonlinear and the classical statistical linear modelling approach, an Autoregressive Integrated Moving Average Model (ARIMA) was constructed using the surge time series data. For a prediction horizon of one hour, the ARIMA model gives not significantly worse performance compared with both LLMs and NN models due to the deterministic persistency and continuity of the dynamical system. However, with the increase of the prediction horizon, it becomes obvious that the underlying dynamics of the system is far from linear. More results and discussions are presented in Chapter 6.

### *3.3.8 Multivariate embedding (input-output systems)*

Up to this point of the discussion we have considered reconstruction of the phase space of the system and its underlying dynamics for time series of single dynamic variable. In principle, because of the embedding theorems that assume infinite long and noise free time series, scalar time series that observe the response of the system are generically sufficient to reconstruct the dynamics of the system provided that enough state variables (delay coordinates) are used. In practice, however, if one has several time series simultaneously measuring different observables with different physical meaning, an alternative approach can be taken for multivariate reconstruction of the phase space. For example, in the thermal convection experiment, time series of measurements of the z-variable of the Lorenz system (3.1) cannot reconstruct properly the dynamics of the Lorenz system if there are not long enough and properly sampled, because of the *x-y* symmetry. More natural approach is to reconstruct the dynamics using all state variables of the Lorenz system. In this special case this would simplify the identification of the dynamics and the analysis of the results, since these variables span the true phase space. However, when one deals with real-life complex nonlinear dynamical systems, one does not know the exact number of the state variables, which drive the dynamics of the system (which is what we are trying to identify), and does not have time series of measurements of all those variables. Moreover, macroscopically meaningful variables are quite often complicated nonlinear functions of several microscopic variables, that can be furthermore very difficult to measure. Thus, the real phase space of the system may embody an attractor that is more folded than a delay embedding of only one dynamic observable. One can try to reconstruct the phase space using the available multivariate time series of measurements. Since the number of the variables is not enough for a full reconstruction of the phase space, one will have to involve additional delay variables. For example, for reconstruction of a nine-dimensional phase space one could use the observed time series of three variables and their proper time delays. This multivariate reconstruction of the phase space does not include conceptual problems in comparison to the univariate embedding, but makes the algorithms somewhat more complicated technically.

An important issue in multivariate time series analysis, and thus the reconstruction of the phase space of dynamical system, is the *relationships* between the variables. This is because they basically provide a simultaneous time evolution of the crucial dynamic variables. Several authors (e.g. Abrabanel, 1994) characterise the relationships between

the dynamical variables as *non-predictive* and *predictive*. The natural approach in modelling is usually to distinguish between the dynamical variables as dependent and independent, thus conceptualising the dynamical system as an input-output system. Non-predictive relationships between the variables investigate relations and structures between the independent and dependent variables by including future information of the dependent variables. Thus, these kinds of relationships are used to analyse the underlying dynamics and cause-effect repercussions rather than building prediction models. On the contrary, predictive relationships do not include future values of the dependent variables while studying the structure and relations between dependent and independent variables. As a result, such multivariate models can be used to carry out simulations to examine the effects of an impulsive change of one or more variables on others, and further as prediction models.

Similarly to the univariate embedding, the reconstruction of the phase space using multivariate time delay embedding procedure requires proper selection of the time delays and embedding dimensions. In mathematical terms this can be expressed as follows: One considers and *M*-dimensional noisy time series each containing *N* measurements $\boldsymbol{s}_1$, $\boldsymbol{s}_2$,…, $\boldsymbol{s}_N$ where $\boldsymbol{s}_i=(s_{1,i}, s_{2,i},\ldots, s_{M,i})$, $i$=1,2,…$N$. As in the case of univariate time series ($M$=1), the time delay reconstruction can be written as:

$$\begin{aligned}\boldsymbol{V}_n = (&s_{1,n}, s_{1,n-\tau_1},\ldots, s_{1,n-(m_1-1)\tau_1},\\ &s_{2,n}, s_{2,n-\tau_2},\ldots, s_{2,n-(m_2-1)\tau_2},\\ &\cdots,\\ &s_{M,n}, s_{M,n-\tau_M},\ldots, s_{M,n-(m_M-1)\tau_M})\end{aligned} \tag{3.94}$$

where $\tau_i$, $m_i$,=1,2,…$M$ are the time delays and embedding dimensions, respectively. Following the embedding theorem, there exist in the generic case a function $F$: $\boldsymbol{R}^d \rightarrow \boldsymbol{R}^d$ ($d=\sum_M m_i$) that maps the current state of the system into the next state,

$$\boldsymbol{V}_{n+1}=F_n(\boldsymbol{V}_n) \tag{3.95}$$

if the total embedding dimension $d$ is sufficiently large. Equation (3.95) can also be written in equivalent form for practical implementation such as

$$\begin{aligned} s_{1,n+1}&=F_{1,n}(\boldsymbol{V}_n)\\ s_{2,n+1}&=F_{2,n}(\boldsymbol{V}_n)\\ &\cdot\\ &\cdot\\ &\cdot\\ s_{M,n+1}&=F_{M,n}(\boldsymbol{V}_n)\end{aligned} \tag{3.96}$$

We can expect that multivariate embedding (3.94) and multivariate models of type (3.96) will include more information and hence a more accurate description and prediction of the underlying dynamics of the system. Moreover, multivariate phase space reconstruction allows the incorporation in the modelling procedure of both the temporal information containing the time series and the spatial information deriving from

considering several variables of the system, and the delayed information. However, looking at equation (3.94) it is obvious that for small values of the individual embedding dimensions, the reconstructed phase space will contain a large amount of delay vectors and thus great redundancy of data. In terms of the theory of nonlinear dynamics, this means that the cloud of points may easily hide the manifold holding the attractor of the system. Furthermore, the trajectory of the system may not be smooth and clear as much as the trajectory revealed from the univariate reconstruction. Therefore, similary to the univariate reconstruction, the problem of reconstructing the phase space using multivariate times series of observables is that of finding the proper time delays and embedding dimensions, which will unfold the attractor of the dynamical system, so that equations (3.95) and (3.96) hold. Choosing the time delays for the multivariate embedding is similar as the choice of time delay from a scalar time series; thus methods and techniques discussed earlier in this chapter hold. Furthermore, one could use for example the mutual information technique between two or three time series of observables to find the optimal time delays.

The choice of the embedding dimensions from multivariate time series data is more difficult and still an open problem. Recently, Cao et al. (1995, 1997) have proposed a method for choosing the *minimum* embedding dimensions based on the average false nearest neighbours. This method draws on the false nearest neighbourhood technique (described in section 3.3.2). The basic idea is to utilise the continuity of the mapping function $F$ in phase space (3.95) or the functions $F_1,\ldots, F_M$ (3.96) if they exist. For practical implementation it is better to consider the mapping functions $F_1,\ldots, F_M$ separately since they may differ (e.g. for the Lorenz system: $\dot{x}$ depends only on $y$ and $x$ itself but not on $z$, while $\dot{y}$ depends on all variables $x$, $y$ and $z$). Thus one could consider, for example, the problem of finding the minimum embedding dimensions of $F_1$ in (3.96) for already chosen time delay $\tau_i$. For any given set of dimensions one could construct series of the delay vectors $\boldsymbol{V}_n$, defined in (3.94), where $n = \max_{1\le i\le M}(m_i-1)\tau_1 + 1,\ldots, N$. For each vector $V_n$ one could find its neighbour $\boldsymbol{V}_{\eta(n)}$, such that

$$\eta(n) = \arg\min\left\{\left\|\boldsymbol{V}_n - \boldsymbol{V}_j\right\| :\ j = \max_{1\le i\le M}(m_i-1)\tau_1 + 1,\ldots, N,\ j \ne n\right\} \tag{3.97}$$

where one could use either the Euclidean norm or some other norm. Having found the neighbouring points, the idea is now to calculate the mean one-step prediction error from a simple neighbourhood local predictor (recall the local zeroth order models), and thus, express the error as a function of the embedding dimensions,

$$E(m_i) = \left(\frac{1}{N - J_o + 1}\sum_{n=J_o}^{N}\left|s_{1,n+1} - s_{1,\eta(n)+1}\right|\right),\quad J_o = \max_{1\le i\le M}(m_i-1)\tau_1 + 1 \tag{3.98}$$

One could also consider different error measures, such as mean absolute error or root mean squared error. Different error measures that are used to assess the quality of the predictions throughout this work are presented in Appendix A. Is obvious that in order to minimise the error (3.99) one needs to solve an optimisation (error minimisation) problem, i.e.

$$(m_{i\,emb}) = \arg\min\left\{ E(m_i) : (m_i) \in Z^M, \sum_{i=1}^{M} m_i \neq 0 \right\} \tag{3.99}$$

where $Z$ denotes all non-negative integers. In this way one could find the minimum embedding dimensions, which minimise the one-step ahead prediction error. However, this does not imply optimal embedding dimensions which will smoothly unfold the attractor of the system. The main drawback of this method is that it is a model based on predefined local models (one must assume the local mapping functions and the number of neighbours) and is very sensitive to noise.

In this work we propose a systematic approach to the reconstruction of the phase space from the multivariate time series. The main idea is to make a compromised use of both, finding a proper multivariate embedding that will ultimately unfold and expose the attractor and dynamics of the system, based on its geometrical and dynamical invariant quantifiers (such Poincare sections, dimensions, entropies and Lyapunov exponents), and finding a suitable embedding seen as a modelling problem (the approach described above). A step by step description of the proposed procedure is as follows.

1. Estimate the time delays and embedding dimensions for all times series of observables separately using the methods already described in this section;
2. Reconstruct the phase space as defined in (3.94);
3. Compute the Poincare sections, information, correlation and Lyapunov dimension of the attractor, and the Lyapunov spectrum;
4. Apply the singular value decomposition (SVD) and the average mutual information (AMI) in order to investigate the redundancy between the delayed vectors;
5. Reduce the redundancy by choosing smaller embedding dimensions (if necessary);
6. Repeat steps 2–5 until there is no significant change in the statistics of estimations of correlation dimension of the attractor and the maximum Lyapunov exponent in particular;
7. Vary the initially estimated time delays in order to investigate the sensitivity of the correlation dimension and maximum Lyapunov exponent estimation. If necessary correct the time delays and repeat the whole procedure starting from step 2;
8. Use the newly (or initially) estimated time delays to find the minimum embedding dimensions following the average false nearest neighbours prediction procedure described above. Investigate the selection of the number of neighbours for the zeroth order local approximation for the variable which is the target of the predictions. If the values of the embedding dimensions differ significantly from the values estimated after at the end of step 7, adjust them and repeat the whole procedure starting at step 2.

The described procedure was initially applied on the Lorenz system. The time series for $x$, $y$, and $z$ variables, each with a length of 20000 samples (time step $\Delta t$=0.01 sec), were generated by numerical integration of the Lorenz system as described in Example 3.1. The noise introduced in the time series due to the truncation error and the nonlinear analysis is estimated to be less than 0.5%. The time delays and the embedding dimensions were obtained separately for each times series and are summarised in Table 3.5.

Table 3.5. Time delays, embedding dimensions and maximal Lyapunov exponents for the Lorenz system variables *x*, *y*, and *z*.

| Variable | $\tau$ | $m$ | $\lambda_1$ |
|---|---|---|---|
| *x* | 18 | 3 | 0.026 |
| *y* | 18 | 3 | 0.024 |
| *z* | 15 | 3 | 0.021 |

Initial multivriate reconstruction of the phase space as defined in (3.94), leads to the series of state vectors each having nine components, composed of the three variables and their time delays,

$$\boldsymbol{V}n=(x_n, x_{n-18}, x_{n-36}, yn, y_{n-18}, y_{n-36}, z_n, z_{n-15}, z_{n-30}) \tag{3.100}$$

Using SVD analysis (described in section 3.3.1), one was able to find that three eigenvalues out of nine are insignificant and their relative constribution is smaller than 1% level. These findings were also supported by the AMIs estimated between the mutual pairs of the vector components. This analysis led to a reduction of the components of the state vectors from nine to six, such as

$$\boldsymbol{V}_n=(x_n, x_{n-18}, x_{n-36}, y_n, y_{n-18}, y_{n-36}, z_n, z_{n-15}, z_{n-30}) \tag{3.101}$$

meaning the reduction of the embedding dimensions to $m_1=2$ for each variable. Investigation of the reconstructed phase space on different values of the time embedding showed that the values between $\tau_i=14\div20$, do not significantly change the reconstruction and for simplicity the time delay for all variables was set to $\tau_i=15$. An example of the 3D representation of the reconsrtructed phase space as defined by (3.101) is presented in figure 3.34.

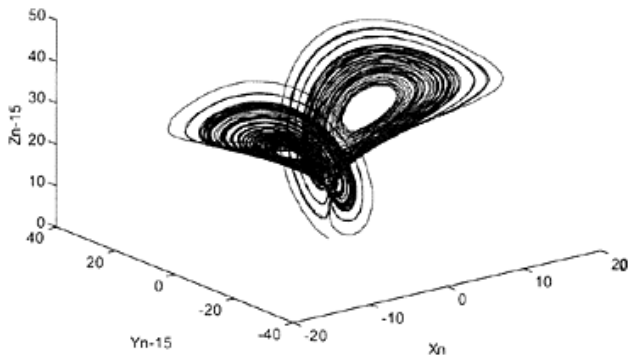



Figure 3.34. 3D view of the mulitvariate reconstruction of the

Lorenz atractor as defined by (3.101) projected to the components ($x_n$, $y_{n-15}$, $z_{n-15}$).

Furthermore, step 8 of the described procedure was applied in order to determine the minimum embedding dimensions for predictive purpuses. Two multivariate zeroth order local predictors were considered, namely

$$x_{n+1} = F_x(x_n, x_{n-15}, \ldots, x_{n-(m_x-1)15}, y_n, y_{n-15}, \ldots, y_{n-(m_y-1)15}), \quad m_x, m_y = 1, \ldots, 20 \tag{3.102}$$

and

$$z_{n+1} = F_z(x_n, x_{n-15}, \ldots, x_{n-(m_x-1)15}, y_n, y_{n-15}, \ldots, y_{n-(m_y-1)15}, z_n, z_{n-15}, \ldots, z_{n-(m_z-1)15}) \tag{3.103}$$

The optimal number of neighbouring points (states) in phase space was found to be $k$=45. By minimising the mean squared error for one-step prediction, the values of the embedding dimensions were estimated as $m_x$=$m_y$=2 and $m_z$=1. These results indicated that a proper multivariate reconstruction of the phase space can be successfully achieved if one uses the time delays of $\tau_i$=15 and embedding dimensions $m_i$=2. However, if the ultimate goal of the multivariate embedding is the prediction of the variable $z$, then using an embedding dimension of $m_z$=1 gives better forecasting results. The multivariate embedding was further used to forecast the values of the variables $x$ and $z$ using zeroth order local models as an average of the images of 45 neighbours. The first 19500 points were used for the phase space reconstruction and the search for the neighbours and the last 500 points of the time series were used to assess the quality of the predictions. The prediction horizon was chosen to be 20 steps ahead. The quality of the predictions was assessed by evaluating the normalised mean squared error (that is the MSE divided by the variance of the test data) and is presented in Table 3.6. The predicted values of the variable $x$ using multivariate zeroth order local models are presented in Figure 3.35 and we also compared with the prediction from a global multivariate neural network model, designed as a three-layered feedforward (architecture 6×9×1) network using static backpropagation learning algorithm.

Table 3.6. The normalised mean squared error for mulivariable zeroth order local predictors for the test data set (500 samples) using prediction horizon of 20 steps for the Lorenz system variables *x* and *z*. The predictions are also compared to a global multivariate neural network model.

| Variable | normalised mean squared error | | |
|---|---|---|---|
| | multivariate zeroth local model | global neural network | univariate zeroth local model |
| *x* (equaton 3.102) | 0.00100 | 0.00144 | 0.00168 |
| *z* (equaton 3.103) | 0.00168 | 0.00201 | 0.00228 |

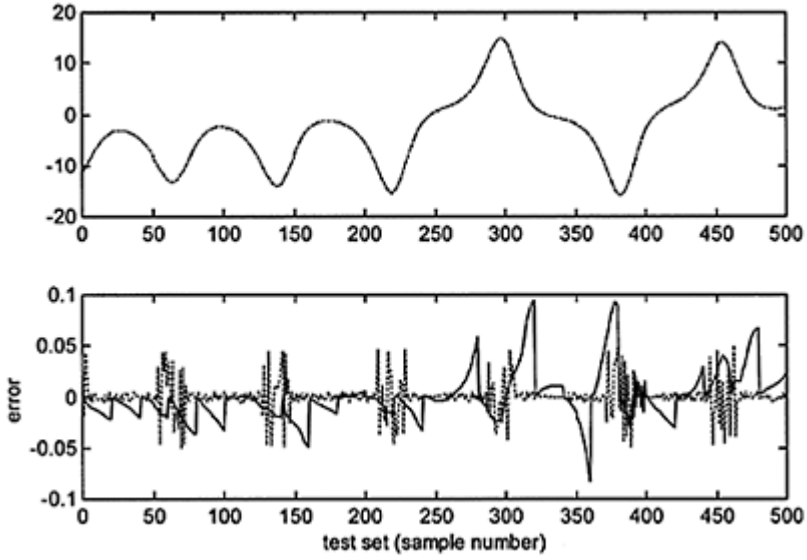



Figure 3.35. Predicted value of the *x* variable of the Lorenz system for predicton horison of 20 time steps ahead. The solid line represents the measured value, the dash-dotted line represent the prediction obtained using the multivariate zeroth local predictors

(3.102) and the dashed line represents the prediction obtained by multivariate neural network (global model). The lower graph represents the errors from the predictions. The solid line is the error between the measurements and the LLMs predictors and the dashed line is the error between the measurements and the NN predictor.

From the results presented in Table 3.6 and Figure 3.35 one can see that the overall predictive capabilities of the multivariate local models are better in this case compared to both the neural network and the univariate local models. The errors are quite small due to the fact that the chosen prediction horizon (20 steps ahead) is still well into the zone of the predictability defined by the largest Lyapunov exponents, which in this case is a maximum of 40–50 time steps in the future (the inverse of the maximum Lyapunov exponents). What is interesting to notice is the fact that the zeroth approximation of the local models (locally constant dynamics) for short-term prediction horizons performs better then a global sophisticated neural network model. The plot of the errors also shows that the locally constant dynamics fails to capture the transitions of the oscillations from the one wing of the Lorez attractor to the other, due to the highly nonlinear effect of these transitions, and thus, exhibits higher absolute errors in comparison to the nonlinear neural network model. A better approach would possibly be to use local linear or maybe second order local polynomials. In Chapter 5 latter in the thesis we will again address the issue of modelling of those nonlinear transitions. Finally, we would like to mention that the computational time for learning the local multivariate models for the setup we used in this experiemnt, took about 20 seconds, while the neural network model took about 1 hour using 10000 training epohs. Practical applications of multivariate local modelling approach are presented in Chaper 6.

## 3.4 Selected nonlinear phenomena

In the preceding sections we have already discussed the properties of deterministic chaotic dynamics, its identification, delineation and modelling, but nonlinear dynamical systems possess much richer phenomenology than just deterministic chaos. Therefore, in this final section we shall address some selected nonlinear phenomena, such as bifurcations, intermittency, spatially extended systems and coexistence of attractors, related to the issues of nonlinear time series analysis and modelling of such dynamical systems.

### *3.4.1 Bifurcations and routes to chaos*

We have already briefly introduced the notion of bifurcation at the beginning of this chapter in the context of stability analysis. Bifurcation occurs where solutions of a nonlinear system change its qualitative character due to a change of parameters and thus some dynamical variables. In particular, bifurcation theory is about how the number of asymptotically approaching steady solutions depends on the parameters of the dynamical systems under study. A bifurcation, contradicting the Linnaeus' assertion that "Nature does not proceed with jumps", may confound intuition, so that many applications of bifurcation theory might be of importance in practice. Typical examples in hydraulics are hydraulic jumps, breaking waves and other flow discontinuities, such as rolewaves. In terms of phase space representation, this implies the change in stability of existing geometrical objects (attractors), but also the birth and death of objects. In this section we shall briefly discuss the most relevant types of bifurcations, which are graphically illustrated in Figure 3.36.

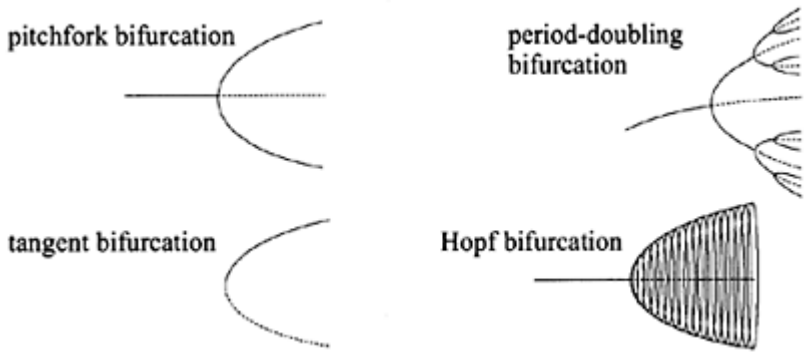



Figure 3.36. Illustrative presentation of different bifurcation diagrams. The abscissa represents some control parameter and the ordinate shows the position of the attractors. Solid lines denote stable orbits while dotted lines unstable orbits. (Figure courtesy of Drazin, 1992).

The *period doubling bifurcation* has been observed in almost any mathematical model system and in many real-life oscillating dynamical systems. At the crucial parameter value a stable period-*p* orbit becomes unstable, and at the same instant a period-*2p* orbit emerges from it (recall the example 3.2). In Fourier language, one prefers the formulation that subharmonics are created in the system. Period doubling bifurcations often appear in a cascade, which is usually the so-called Feigenboum scenario, meaning that further period doublings occur at higher values of the control parameter, until at some stage infinite period is reached and the motion becomes chaotic (recall the broad-band spectra).

This type of bifurcations exhibit many universal details (see, for example, Ott(1993)), from which the most interesting one is the scaling law between the parameter values $r_p$, $p$th period and the distance towards the critical value of the parameter, such as $r_p - r_c \propto \delta^p$.

Very similar in its structure and bifurcation diagram is the *pitchfork bifurcation*. In this case a stable fixed point becomes unstable and two new stable fixed points are created, and in fact, this kind of bifurcation may usually take a part of the period doubling bifurcation, as shown in Figure 3.36.

The *tangent bifurcation* (saddle-node bifurcation) is a very characteristic bifurcation, and its mechanism describes typically how, inside a chaotic dynamical regime, with a change of the control parameter a stable periodic solution may suddenly occur.

Finally, the last bifurcation we want to mention is the *super-critical Hopf bifurcation*. For a value of a certain control parameter where a stable fixed point becomes unstable a stable limit cycle is born. Furthermore, a limit cycle can itself bifurcate into a two-frequency torus by another Hopf bifurcation. Under further changes of the system parameters the torus becomes unstable and may either develop into a hyper-torus via a Hopf bifurcation or even in a chaotic attractor (with fractal dimension). This is the route to chaos by quasi-periodicity and is very common for hydrodynamical systems. In addition, many hydrodynamical systems, and in particular the externally driven ones, exhibit *symmetry breaking bifurcations*.

All of the above bifurcations are continuous in the sense that the response to small dynamical perturbations due to the change in system parameters is still small, but growing continuously exponentially. In contrast, a sudden change of chaotic attractors may occur, which is usually called a *crisis*. It is a very fast occurring global bifurcation and thus difficult to study from a mathematical point of view. A crisis may manifests itself in a sudden disappearance of the attractor of the dynamical system, in a sudden change of its size or the merging of two attractors. In nature one may speak of natural disasters, such as flash floods, fast forming storm surges, spontaneous atmospheric pressure fields etc.

As an illustrative example of the mechanisms inherent in the different types of bifurcation discussed and in the route to chaos, we present in Figure 3.37 the bifurcation diagram of the very simple logistic map $x_{n+1}=\mu x_n(1-x_n)$, for different values of its controlling parameter μ. It is evident that the period doubling and pitchfork bifurcations occur for a while. The map has a cycle 2 for example when the controlling parameter $\mu=3.4$. This cycle further bifurcates to cycles of period 4,8,16, etc., as μ increases. Above the value of the control parameter $\mu\approx3.57$, the map exhibits deterministic chaos, but once the chaotic dynamics is achieved, by further increase of μ there are evident appearances of zones, where the periodic motion has established again (tangent bifurcations).

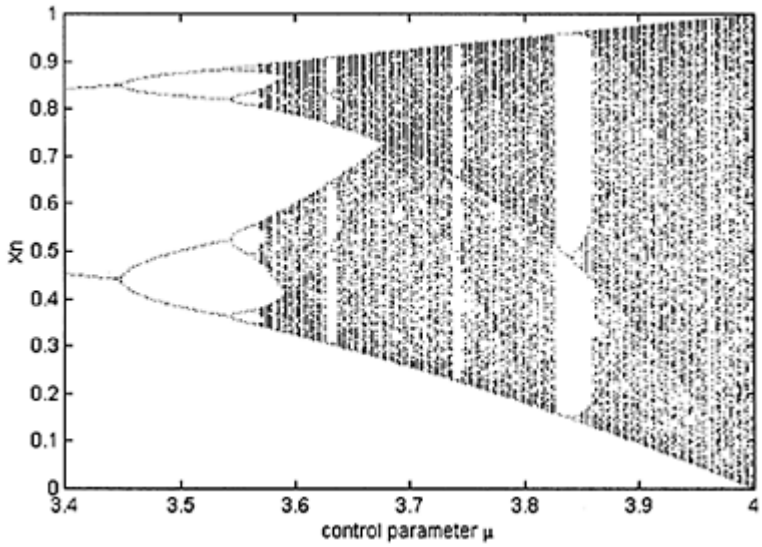



Figure 3.37. The complete bifurcation diagram of the logistic map.

This example shows that even a very simple nonlinear dynamical system may generate a reach dynamical structure based on slight changes in its control parameters. Obviously, in order to study bifurcations with a time series of observables, one has to be able to obtain a time series that embodies significant information regarding the response of the system under different dynamical regimes, and under different physical conditions, and even multi-channel measurements. Typically, candidates of such dynamical systems could be numerical models. For in-depth study of the theory of bifurcation we recommend the textbooks of Drazin (1992) and Ott (1993). Furthermore, in Chaper 6 we will present a practical application of the bifurcation mechanisms and the route to chaos using time series obtained by three-dimensional hydrodynamical model simulations.

### *3.4.2 Intermittency*

Another group of interesting nonlinear phenomena is known as *intermittency*. The notion of itermittency means that the dynamical evolution of the system alternates between periodic (regular, smooth, laminar) and chaotic (irregular, turbulent) behaviour in an irregular manner. The chaotic phases can occur as long bursts or they can look like short bursts, depending on the dynamical properties of the system. For example, in spatially extended hydrodynamical systems, such as weekly turbulent flow, intermittency means a simultaneously appearance of turbulent and laminar phases at different places in real space. Several different scenarious have been proposed in the literature (see Bergé et al., 1986) to study and explain intermittency in deterministic dynamical systems. As reported by Berge at al., 1980 in some deterministic systems, such as thermodynamical and hydrodynamical systems, intermittency occurs generically. In other systems, it occurs

only for critical values of some control parameters (Pomeau and Manneville 1980). In such systems, periodic orbits may become unstable under the change of some control parameters or it may disappear without creation of new ones. In order to be able to study the nonlinear mechanisms of intermittency from a time series, one can compute the statistics of this type of dynamics, i.e. the distribution of the inter-burst times and the duration of the bursts, and, furthermore, one can try to establish relations between the average length of periodic phases and the closeness of the parameter to its critical value. This implies that one needs to be able to obtain detailed time series from controlled experiments.

### *3.4.3 Spatialy extended systems*

Up to this point of presentation and discussion, we have assumed that nonlinear deterministic dynamics, responsible for the observed time series, dominantly possess attractors of low dimensions (dimension less than the essential degree of freedom) that can be estimated and regarded as flows. However, many dynamical systems, and especially hydrodynamical systems, are composed of a huge number of microscopic degrees of freedom, which mathematically are usually formulated by systems of partial differential equations. Therefore, their degrees of freedom are attached to the points in space and can be interpreted as local amplitudes, imposing that the spatially extended dynamical systems exhibit infinite degrees of freedom, meaning that the complete (true) phase space cannot be even reconstructed. Generally speaking, in almost all hydrodinamical systems coupling between the different degrees is diffusive. For example, in one-spatial dimension and with arbitrary forcing (driving) term $q(x, t)$, the dynamics of such systems can be generally described by partial differential equations of type

$$\frac{\partial u}{\partial t} - D\frac{\partial^2 u}{\partial x^2} + F(u, \frac{\partial u}{\partial x}) = q(x,t) \qquad (3.104)$$

where the second term describes the diffusive coupling and the third term describes the nonlinearities. Of course, the most prominent example in hydrodynamics are the Navier-Stokes equations, where $\mathbf{u}(\mathbf{x}, t)$ is the vector valued local velocity field of the fluid, the term $F(u, \partial u/\partial x)$ has the nonlinear form $\mathbf{u}(\nabla\mathbf{u})$, and the driving term is the gradient of the pressure. The usual approach to solve the dynamics of such complex spatially extended systems is utilising various numerical methods developed in computational hydraulics and mechanics, by discretisation of the space and application of certain numerical schemes (approximations) to the partial derivatives in those points (see, for example, Abbot, 1986). Discretisation of the space leads to a system of an enumerable set of degrees of freedom, $u(x, t) \approx u(x_i, t)$, and in order to avoid integration step one often discretises time also. Such resulting approxime dynamical systems of difference (mapping) functions in the theory of nonlinear dynamical system are called *coupled map lattices*. In each lattice site $i=1,2,\ldots,L$, the dynamics is expressed as, for example:

$$u_{i,n+1} = (1-2\varepsilon)F(u_{i,n}) + \varepsilon(F(u_{i-1,n}) + F(u_{i+1,n})) \qquad (3.105)$$

where the first index in $u_{i,n}$ is the spatial index, the second index is the temporal one, and ε is some indicator based on the neighbourhood. The nearest neighbour coupling used in the above expression is again diffusive, since it is the simplest representation of the Laplacian on the lattice. An even further step of discretisation would involve discretisation of the possible states of the system. This means that if for the individual degrees of freedom, $u_i$, one could assume only a finite number of different values (in simplest case 0 and 1), then one can speak of a *cellular automation* or *cellular automata* (CA). Cellular automata have five major characteristics:

1. they exist on lattices of discrete sites;
2. the time evolution takes place in discrete steps;
3. each site has a finite number of possible values (states);
4. each site evolves in time according to a deterministic rule (mathematically known);
5. the rule for each site depends only on some finite neighbourhood on the lattice and a finite number of previous time steps (e.g. rule 3.105).

In spite of their simplicity cellular automata show a surprising ability to reproduce complex dynamics. Applications of cellular automata can be found in non-equilibrium physics, population dynamics, chemical reactions, epidemiology, parallel computing, geophysics and hydrodynamics (see, for example, Wolfram, 2002; Wilson 1988 for introduction and Abbot and Minns, 1997 for a general discussion of the application of CA to turbulence modelling). Due to their discrete nature CA are particularly well suited for implementation and simulation on digital computers. However, although they look simple, CA implementation in real application problems may demand a huge computational capacity. For example, consider simple CA where automation sites are arranged on a one-dimensional lattice. All sites are identical, they may have a finite number $k$ of possible values, and there is a dependence only on the previous time step. Furthermore, let the automation (update of the values at each site) depends only on the value of the site itself and its two immediately adjacent sites in the previous time step. Graphically, the structure of this CA can be represented as a simple discretised line with three highlighted sites (points). Thus there are $k^3$ possible initial "local" states at any given site on the lattice. Since there are k possible results (values), the total number of possible rules is $k^{k^3}$. In the simplest case when $k$=2 (only two possible values, e.g. 0 and 1) there are 256 possible rules. It is obvious that this number increases extremely rapidly with increasing $k$. Thus $k$=3 gives about $7.6\times10^{12}$ possible rules that have to be searched through at each updating time step for this very simple example.

During the spatial evolution of the dynamics if the coupling is diffusive, this part of the dynamics cannot introduce irregularities. Diffusion smooths down all excitation patterns, and usually the diffusion operator (the Laplacian) has only negative eigenvalues, which contribute to exponentially decaying solutions. Therefore, only the nonlinearities (which in equations (3.104) and (3.105) are purely local) can cause deterministic chaotic solutions of the underlying dynamics. The two effects can lead to what is usually called *spatiotemporal chaos*. It is characterised by an instability and exponentially decaying correlations both in time and space directions (Toricini et al., 1991). This means that local perturbations spread in space with increasing instead of decreasing amplitudes in contrast to global diffusion processes.

## ANALYSIS OF SPATIALLY EXTENDED SYSTEMS

One can further pose the question how to analyse these kind of spatially extended dynamical systems from a time series of observables using nonlinear methods. In many situations it will be most convenient to focus on a particular spatial part (subsystem of interest) and to measure similar physical quantities simultaneously at different positions in this part of the dynamical systems. Using these time series of spatially distributed measurements one can try to reconstruct the phase space of the system in a spatial direction, by choosing the embedding dimension and spatial distance $\delta$ between the positions of the measurements. The latter again attempts to find the best compromise between redundancy and decorrelation. As for the time delay, the mutual information between two simultaneously measured variables as a function of their spatial distance can be used. It has been conjectured and demonstrated for number of mathematical models that systems exhibiting spatiotemporal chaos have attractors which are not finite-dimensional but possess a finite dimension-density (Bauer at al., 1993, Tsimring, 1993, Torcini et al., 1992). This means that the dimension determined for a subsystem is proportional to its spatial extent. Spatially extended systems can be characterised by Lyapunov spectra, entropies and dimensions, just like low dimensional deterministic chaos. The unknown, however, lies in the dependence of these geometrical and dynamical quantities on the spatial scales. It has been proven by Ruelle (1982) and others for partial differential equations and showen numerically for many coupled map dynamical systems (see, for example, Paladin and Vulpiani, 1986), tat the spectrum of Lyapunov exponents has, in the limit of the large spatial scales $L$, the following property:

$$\lambda_i^{(L)} = \lambda(i/L) \tag{3.106}$$

In other words, this means that the $i$-th Lyapunov exponent in a system with size $L$ has a value which depends on the ratio $i/L$ only. Under the assumption that the Pensin identity (3.54) holds, the Kolmogorov-Sinai entropy can be approximated from the Lyapunov spectrum, such as $h_{KS}^{(L)} = \sum_{i\lambda_i>0}\lambda_i \approx \int_0^{x_0}\lambda(x)\mathrm{d}x$, where $\lambda(x_0)=0$, such that the entropy of the system is proportional to its spatial scale. Similarly, by applying the Kaplan-Yorke relation (3.45), one can estimate the Lyapunov dimension proportional to the spatial scale of the system.

Dimension, entropy and Lyapunov densities can be estimated from a time series obtained by numerical simulation of spatially extended dynamical systems. It is, however, still an open issue whether it is possible to determine these densities from the time series of observables on a small part of the system. Even for a multivariate time series which represents a finite subsystem of an extended system conclusions and extrapolations of the results are difficult. Thus, in order to study spatiotemporal nonlinear dynamics from time series properly, one needs a vast amount of multichanneled and multivariate time series data sampled sufficiently over the spatial domain of interest. On the other hand, studying in detail the complete spatial dynamics and evolution of the complete states of the system in microscopic level may not be of practical interest. In practice, one is more interested in studying the so-called *spatiotemporal patterns* such as, for example in hydrodynamical systems, rotating spiral waves, appearance propagation and deformation of large eddies, moving fronts, breaking waves etc. The patterns

mentioned above have a clear signature in the real space. Once they are recognised, based on their characteristics e.g. velocities, geometries, length scales, then one can proceed to a deeper study of their emergence (pattern formation), evolution and interaction (see Cross and Hohenberg, 1993). Utilising the sophisticated remote image sensing techniques available nowadays, one is able to record the dynamical evolution of spatially extended systems, in a sense that values of particular variable are assigned to every small element of the grid (pixel) over time. Image analysis techniques, based on pattern recognition, together with some of the techniques we described in this chapter, such as wavelet decomposition, singular spectrum analysis and fractal dimension estimation techniques, can be employed to study these spatiotemporal patterns.

Finally, we would like to stress the importance of local modelling in phase space in the analysis of spatially extended dynamical systems based on time series data. All numerical engines that at present are used to approximate the dynamics of the spatially extended systems, usually described mathematically by a set of partial differential equations based on physical laws, exploit neighbourhoods in some way or another by the numerical methods employed (e.g. equation 3.105). Thus, these are in general set of local mapping functions (algebraically expressible) which map the evolution of the states of the system throughout the *domain of dependency*. On the other hand, all methods and techniques discussed in this chapter that are used to reconstruct, delineate and model the dynamics of the system by nonlinear time series analysis, are also almost exclusively based on neighbourhood statistics. The dimension is, loosely speaking, the rate at which the point on the attractor loses its neighbours if one decreases the radius of the neighbourhood. The entropy correspondingly is the information loss rate if one increases the embedding dimension at a fixed length scale. The maximal Lyapunov exponent quantifies the loss of information with the increase of distances between neighbours. Building models in phase space, learned from data, are parametric mapping functions (algebraically expressible) that describe the evolution of the system states based on the previous evolutions of their neighbours. Based on the discussion above, one could infer the following conclusion: Learning parametric local models (e.g. linear or polynomials) in the phase space reconstructed from a time series of observables of spatially extended dynamical system, is conceptually equivalent to applying a certain numerical scheme (e.g. finite difference) as approximate numerical solution in physically-based mathematical models.

### *3.4.4 Multiply or coexisting attractors*

One remarkable feature that can occur in nonlinear dynamical systems is the coexistence of attractors or basins of attractions. The attractor towards which the dynamics of the system will evolve depends on the initial and/or boundary conditions and the values of the control parameters. Theoretically this means that the long time evolution of the dynamical system exhibits several trajectories or even interchange between them given certain physical conditions. The coexistence of the attractors can be theoretically studied on mathematically defined dynamical systems by reconstruction of the phase space for different evolutions of the system using different initial and/or boundary conditions and sets of control parameters. Such an example we have already described and demonstrated in Section 3.1 (recall example 3.3) using the well-known predator-prey dynamical

system. The region of phase space leading to a given attractor is called a *basin of attraction*. The basins of two coexisting attractors can even interlace in a very complex way, possessing fractal boundaries (Grebogi et al., 1983). The simplest coexisting attractors are, for example, two different stable points or limit cycles which are often related by some symmetry. A typical example could be a groundwater hydrodynamical system with two wells. Another example is again the famous Lorenz dynamical system. For certain ranges of the values of the parameters and initial conditions there coexist stable fixed points and fractal attractors. Furthermore, the strange attractor of the system is composed by two geometrical objects (two wings), which themselves constitute strange attractors and parts of the trajectory that link them. The evolution of the system alternates between these two parts of the attractor (subattractors), clearly revealing the existence of two dynamical regimes.

Most natural dynamical system, for example hydrometeorological systems, show a similar type of behaviour. The studies of weather patterns using nonlinear methods have demonstrated the coexistence of several attractors in the weather systems on different time scales (see, for example, Elsner and Tsonis, 1988 and 1992). The historical records of the outputs of most hydrological systems, for example runoff of a particular catchment, in general show presence of several distinguishing parts in the evolution due to the presence of seasonal components, normal and extreme events. Moreover, in most natural dynamical systems one could even observe a nonunique mapping between different dynamical variables of the system. A typical example is the histeresis-type of water level—discharge relationship during flooding events. Reflected in terms of nonlinear dynamics and phase space reconstruction of the system, this indicates possible existence of distinguishable different geometrical parts in the attractor or even coexisting attractors, which are characterised by different underlying local dynamics (dynamical regimes) and transition between them driven by the physical conditions of the system and the interacting systems. Obviously, reconstruction of the phase space of such complex dynamical systems from time series of observables, and in particular identification of the existence or nonexistence of different dynamical regimes, requires time series with proper time and length scales that will in turn reveal the complicated geometrical structure of attractor(s) and underlying global dynamics. Form a modelling perspective, one could think of exploring the possibility of hybrid modelling of the complex global dynamics, consisting of: (i) models with different structures and capacities which will be learned and specialised to map the local dynamics in phase space, thus modelling different dynamical regimes; and (ii) a gating model in phase space which learns to map the transitions (specific parts of the trajectories in phase space) based on the previous evolution of the state variables and different dynamical conditions. This idea of hybrid modelling in phase space of complex nonlinear dynamical systems is further explored and discussed in Chapter 5 of this work, where we propose and elaborate a novel hybrid modelling framework.

## 3.5 Summary

The paradigm of nonlinear dynamics and the concept of deterministic chaos in the last decade have influenced the thinking and problem solving in many fields of science and engineering. We showed that as models chaotic dynamical systems show rich and even surprising variety of dynamical structures and solutions. Most appealing is the fact that the deterministic chaos provides a prominent explanation for irregular behaviour and instabilities in dynamical systems, which are deterministic in nature. The most direct approach to reconstruct, identify, quantify, model and control deterministic chaos in real dynamical systems is the analysis of data (time series) generated from these systems using methods and techniquies based on the theory of nonlinear dynamics, which we elaborated and demonstrated in this chapter.

# Chapter 4
# Dynamic Bayesian Networks

## 4.1 General

In Chapter 2 we have briefly introduced the Bayesian approach to learning. We have also discussed that the Bayes chain rule provides a natural mathematical framework for learning and inference where background domain knowledge can be incorporated in the learning procedure. Furthermore, we also demonstrated that the Bayesian approach to learning inherently embodies the structural risk minimisation principle, thus providing a sound framework for regularisation and generalisation. In this chapter we further extend this discussion and present a probabilistic framework for learning models from time series data. We express these models using the Bayesian network formalism (also known as probabilistic graphical models or belief networks), that is, a marriage between probability theory and graph theory in which dependencies between variables are expressed graphically. Special attention and focus will be given to dynamic Bayesian networks, which are well suited for learning models from time series data observed from complex dynamical systems.

There are at least several reasons for elaborating dynamic Bayesian networks and Bayesian methods in this work. *Firstly*, dynamic Bayesian networks can handle incomplete data sets, in a sense that not all of the dynamical variables needed to completely describe the dynamical evolution of the system are observed as a time series and thus some of the variables can be regarded as hidden in the processes of inference and learning. *Secondly*, dynamic Bayesian networks, expressed as graphical models, allow to learn and understand the causal relationships between the variables. The process is useful when one tries to gain understanding about the problem domain based on the time series data, for example, during the exploratory data analysis. In addition, modelling the causal relationships allows one to make simulations of the dynamical systems in the presence of interventions. *Thirdly*, dynamic Bayesian networks in combination with Bayesian methods allow for an efficient computation of marginal and conditional probabilities that are required in the inference and learning procedures. Moreover, the inherent regularisation capabilities provide principled approaches for avoiding overfitting the data, model selection and model averaging. *Fourthly*, as already mentioned, Bayesian methods can facilitate the combination of domain knowledge and data. Anyone who has performed real-world modelling tasks knows the importance of prior domain knowledge, especially when the data is scarce and incomplete. In addition, Bayesian networks in general can be built from prior knowledge alone, thus representing expert (knowledge-based) systems, which are well-known to the artificial intelligence community. The causal semantics present in Bayesian networks makes the encoding of causal prior knowledge particularly straightforward. The way of encoding the strength of causal relationships in Bayesian networks is by use of probabilities. *Fifthly*, various

uncertainties (such as model and parameter uncertainties, uncertainties in the domain knowledge) in Bayesian networks are handled in probabilistic manner. *Finally*, dynamic Bayesian networks in conjunction with Bayesian methods provide a framework for richer hybrid models appropriate for a time series describing the dynamics of complex and multiresolution systems. In the previous Chapter 3 we have elaborated and demonstrated that nonlinear dynamical systems show very rich and complex dynamical structures in their evolutions, characterised by the presence of chaotic dynamics, different dynamical regimes (even coexisting attractors) and an irregular dynamical evolution between them. These irregular dynamical transitions between different dynamical regimes (or modes) of the system can be modelled using dynamic Bayesian networks based on the information of the position in the phase space, previous evolutions, and performance of the models that are particularly learned and suited to model different parts of the dynamics of the system. This framework will be discussed and mathematically elaborated in the next Chapter 5.

Having stressed the motivation for the description of dynamic Bayesian networks, we discuss first the Bayesian interpretation of probability and review methods from Bayesian statistics for combining prior knowledge and data. Furthermore, we introduce and describe Bayesian networks in general and focus on dynamic Bayesian networks for modelling time series, including the well-known Kalman filter and Hidden Markov Models (HMMs). In addition, we elaborate on the problem of learning the parameters of dynamic Bayesian networks, especially the presence of hidden variables using the Expectation-Maximisation (EM) algorithm. Computing various probabilities and inference in such models may be computationally intractable, therefore we briefly discuss the use of Monte Carlo approximation techniques and Gibbs sampling in particular. Finally, we discuss some issues related to richer structures of dynamic Bayesian network and their relation to modelling complex nonlinear and non-stationary dynamical systems.

## 4.2 Bayesian approach to probability and statistics (inference)

In order to understand Bayesian networks and learning, it is necessary to briefly introduce the classical and the Bayesian approach to probability, followed by some basic introduction to graph theory.

### *4.2.1 Classical definition of probability*

When using the classical probability, one may think in terms of: probability that a storm will occur; the probability that a particle will not follow certain trajectory; the probability of observing a string of three rainy days in a month. We use here the general term *sample point* to refer to the "things" we are talking about; thus in this particular case, an abstraction of a storm event, a geometrical point, or a chance outcome. A *sample space* $\Omega$, or *universe*, is the set of all possible sample points in the situation of interest. The sample points in a sample space must be mutually exclusive and collectively exhaustive. A probability measure $p(\cdot)$, is a function on *subsets* of a sample space $\Omega$. These subsets are referred to as *events*. The values $p(A)$, $p(B)$, $p(A \cup B)$ are called probabilities of

the respective events (for $A, B \subseteq \Omega$). This classical approach to probability dates back to Laplace (1749–1827) and his contemporaries and remains the most common way of dealing with probability. At its heart we find the Laplace's classical definition of probability (Laplace, 1951):

> The theory of chance consists in reducing all of the events of some kind to a certain number of cases equally possible, that is, so to say, such as we may be equally undecided about in regard to their existence, and in determining the number of cases favorable to the event whose probability is sought. The ratio of this number to that of all the cases possible is the measure of the probability.

One can write the formula for the classical probability as:

$$p(x) = \frac{N(f)}{N(s)} \tag{4.1}$$

where, $N(f)$=number of favourable outcomes

$N(s)$=total number of possible events in the sample space.

Expressed in mathematical terms, we present here the definition of probability (adopted from Neapolitan, 1990).

*Definition 4.1:*
A probability measure on a sample space $\Omega$ is a function mapping subsets of $\Omega$ to the interval [0,1] such that:

1. For each $A \subseteq \Omega$, $p(A) \geq 0$;
2. $p(\Omega)=1$;
3. For any countably infinite collection of *disjoint* subsets of $\Omega$, $A_k$, $k=1,\ldots,$

$$p\left(\bigcup_{k=1}^{\infty} A_k\right) = \sum_{k=1}^{\infty} p(A_k) \tag{4.2}$$

In general, one needs to check that the sets (events) themselves satisfy certain properties to ensure that they are *measurable* and that the probabilities are assigned by the principle of *indifference*. The fundamental idea of this principle is that alternatives are always to be judged equiprobable without any reason to expect or prefer one over the other (Weatherford, 1982). In general, the probabilities can be estimated by counting the realisations or some other form of direct measurement. In this sense, the probability is an attribute or a property of the real world. The term *physical probability* is often used to denote this interpretation of probability (Heckerman, 1997). More details on these classical probability issues can be found in Chung (1979). For an excellent philosophical discussion and historical outlook on the main developments in probability theory we refer the reader to Jaynes (1995, 1996).

Based on these physical probabilities, a branch of the statistics known as inferential (or frequentist) statistics, that attempts to make valid predictions based on only a sample

space of all possible observations was developed. For example, imagine a bag of 10000 marbles of which some are black and some white, but the exact proportion of these two colours is unknown. The inferential statistics implies that it is necessary to count all the marbles in order to make some statements about this proportion. A randomly acquired sample of 1000 marbles may be sufficient to make an inference about the portion of black and white marbles in the entire population. If 40% of this sample consists of white marbles, than one may be able to infer that about 40% of the population are also white. This inference seems rather straightforward. In fact, it might seem that there is no need to even acquire sample of 1000 marbles, but a sample of 100 or even 10 marbles may be sufficient. However, this assumption is not necessarily correct. As the sample size becomes smaller, the potential for error grows. For this reason, inferential statistics has developed numerous techniques for stating the confidence level (certainty) that can be labelled on these inferences.

### *4.2.2 Bayesian definition of probability*

The classical inferential approach (frequentist or objective) based on the classical probability theory does not permit the introduction of prior (background) knowledge into the calculations. For the rigours of the scientific method, this is an appropriate response to prevent the introduction of extraneous data and information that might skew the experimental results. However, there are certainly times, such as learning models from data as an ill-posed problem, when the use of prior knowledge would be a useful, and indeed a necessary, contribution to the modelling process.

In contrast to the classical probability, the Bayesian probability of an event $x$ is a person's *degree of belief* in that event. This interpretation of the probability is called subjective or Bayesian interpretation, in honour of the Tomas Bayes, who helped to pioneer the theory of probabilistic inference. Whereas a classical probability is a physical property of the world, a Bayesian probability is a property of the person who assigns the probability. One important difference between the physical probability and personal probability is that the latter does not require repeated trials. In the Bayesian approach, the probability or belief will always depend on the state of knowledge of the person who assigns that probability. For example, if we were to give someone a coin, he would likely assign a probability of ½ to the event that the coin would show heads in the next toss. If, however, we convinced that person that the coin is special and was weighted in favour of heads, he would assign a higher probability to the event.

One common criticism of the Bayesian definition of probability is that probabilities seem arbitrary. Normally, questions arise: Why should degrees of belief satisfy the rules of probability? On what scale one should measure the probabilities? These questions have been studied expensively by various researchers in the past (e.g., Ramse, 1931; Cox, 1946; Good, 1950; Savage, 1954; DeFinnetti, 1970). Regarding the first question, it turns out that each set of properties leads to the same rules: the rules of probability. The fact that the different sets lead to the rules of probability provides a particularly strong argument for using probability to measure beliefs. At this point of discussion it is worth mentioning that during the last two decades a completely new branch of the set theory was developed, termed *fuzzy set theory* (fuzzy subsets more precisely) or *fuzzy logic* (Zadeh, 1989). The subjective degrees of belief are expressed through fuzzy membership

functions, which are further used in fuzzy reasoning and inference. A more detailed discussion on this topic is however outside of the scope of this thesis.

In regard with the second question, related to the scale in the process of measuring the degree of beliefs (probability assessment), it makes sense to assign a probability of one (or zero) to an event that will (or not) occur, but the major issue is what probabilities one should assign to beliefs that are between the extremes. This question has been extensively studied and discussed in the Operations Research, System Analysis, Management Science and Psychology literature. The main problem with the probability assessment is that of precision. Can one really say that her or his probability of event $x$ is 0.601 or 0.599? In general no, since in most cases the probabilities are used to make decisions, and these should not be very sensitive to small variations in probabilities. The questions of precision and accuracy are elaborated in the well-established practice of *sensitivity analysis* (e.g. Howard and Matheson, 1983; Spetzler et al., 1975). Furthermore, in order to avoid such questions and discussions, some authors have proposed proposing that the probability assessment procedure is seen as an act of *expert judgement* (e.g. Bernardo and Smith, 1994; Krause, 1998). Whichever view one takes, the Bayesian probabilistic approach offers a mechanism by which the probability estimates may be revised in the light of experience and evidence.

Let us return to the issue of learning from data using the Bayesian approach. In order to examine the Bayesian analysis we shall introduce some notation for the mathematical description in this chapter. We shall replace the term events with the variables[1]. We denote a variable by an upper case letters (e.g. $X$, $Y$, $X_i$, $\Theta$). A variable has a set of states corresponding to mutually exclusive and collectively exhaustive set of events, about which we may be uncertain. A variable may be discrete, having a finite or countable number of states or it may be continuous. For example, we may use a two-state or binary variable to represent the possible working regimes of a water pump, and a continuous variable to represent its operational capacity. The state or value of a corresponding variable is denoted by the same letter in lower case (e.g. x, y, $x_i$, $\theta$). A bold notation is used (e.g. $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{X}_i$, $\mathbf{\Theta}$) to denote a set of variables and the corresponding bold lower case letter (e.g. $\mathbf{x}$, $\mathbf{y}$, $\mathbf{x}_i$, $\boldsymbol{\theta}$) to denote an assignment of state or value to each variable in a given set. This is usually known as the variable set $\mathbf{X}$ is in configuration $\mathbf{x}$. We use $P(X=x|\xi)$ (or $P(x|\xi)$) to denote the probability that $X=x$ of a person with state of information $\xi$ (sometimes called background knowledge). In the problems further considered, we define $\Theta$ to be a variable whose values $\theta$ may correspond to the possible true values of the physical probability. We sometimes refer to $\theta$ a *model parameter*. We express the *uncertainty* about $\Theta$ using probability density function $p(\theta|\xi)$. In addition, we use $D=\{X_1=x_1, X_2=x_2,\ldots, X_N=x_N\}$ to denote the data set of our observations.

[1] The term “random variable” is often used in the literature. We shall reserve the term “random variable” for the situations where there are repeated observations, which is not strictly the case for the variables used in Bayesian learning and Bayesian networks as discussed in this chapter. Bayesians sometimes use the term “uncertain variable”.

## CONDITIONAL PROBABILITY

An important concept in the Bayesian treatment of certainties in Bayesian networks and inference is conditional probability. A conditional probability statement is of the following kind:

> "Given the event *B* (and everything else known is irrelevant for *A)*, then the probability of the event A is *r*."

The above statement is denoted as $P(A/B)=r$. This represents the statement "if *B* is true and no other information at hand is relevant to *A*, then the probability of *A* is *r*". If we are counting sample points, we are interested here in the fraction of events *B* for which *A* is also true, thus we are switching our attention from the sample space $\Omega$ to the subset *B*. Conditional probabilities are essential to a fundamental rule of probability calculus, the *product rule*. The product rule defines the probability of a conjunction of events:

$$P(A, B)=P(A|B)P(B) \tag{4.3}$$

where $P(A, B)$ is the probability of the joint event $A \wedge B$ and also called *joint probability* distribution of events *A* and *B*. If all these probabilities are conditioned by a context *C*, then the Eq. (4.3) can be modified as:

$$P(A, B|C)=P(A|B, B, C)P(B|C) \tag{4.4}$$

From the Eq. (4.3), we can deduce the following relation for conditional probability with $P(B)\neq 0$, which is in fact the simplest form of Bayes' theorem:

$$P(A|B)=\frac{P(A,B)}{P(B)} \tag{4.5}$$

## CHAIN RULE OR FACTORISATION

A joint probability distribution over *n* variables can be defined recursively using the product rule (Eq. 4.3) as follows:

$$\begin{aligned} &P(X_1, X_2,\ldots, X_n)=P(X_1|X_2,\ldots, X_n)P(X_2,\ldots, X_n)\\ &= P(X_1|X_2,\ldots, X_n)P(X_2|X_3,\ldots, Xn)P(X_3,\ldots, X_n)\\ &=P(X_1|X_2,\ldots, X_n)P(X_2|X_3,\ldots, X_n)\ldots P(X_{n-1}|X_n)p(X_n) \end{aligned} \tag{4.6}$$

The above Eq. (4.6) can be generalised as:

$$p(X_1, X_2,..., X_n)=\prod_{i=1}^{n} p(X_i|X_1,..., X_{i-1}) \tag{4.7}$$

This property of joint probability distribution is called the general factorisation property. It is noted that the product rule allows any ordering of variables in the factorisation. This property is very important in decomposing the large joint probability distribution, which can be easily transparent and tractable.

## MARGINALISATION

Let us consider a variable $A$ having states $a_1, a_2,\ldots, a_n$, then the probability distribution over these states is given by:

$$P(A)=(x_1,x_2,\ldots,x_n) \quad x_i \geq 0 \quad \sum_{i=1}^{n} x_i = 1 \tag{4.8}$$

where, $x_i$ is the probability of $A$ being in state $a_i$ denoted by $P(a_i)$. Similarly the variable $B$ has states $b_1, b_2,\ldots, b_m$. Note that if $A$ and $B$ are discrete variables, then the joint probability distribution $P(A, B)$ is a table of probabilities for all possible pairs $P(a_i, b_j)$. In the case when the variables $A$ and $B$ are continuous, the underlying probability density functions are required for the joint probability distribution $P(A, B)$. For illustration we assume that the variables A and B are discrete. Then the joint probability distribution table for $P(A, B)$ will be also a table of $n\times m$, which consists of a probability for each configuration $(a_i, b_j)$. If the probability distribution for each state of variable $B$ is known, then we can produce the probability distribution table for $P(A, B)$ by applying the product rule Eq. (4.3) as:

$$P(a_i,b_j)=P(a_i|b_j)P(b_j) \tag{4.9}$$

In order to understand this concept clearly, we present the following example: Table 4.1 consists of probability distribution for $P(A|B)$. The probability distribution for variable $B$ is known (0.4, 0.4, 0.2). Then the joint probability distribution $P(A, B)$ is obtained by multiplying each j the column for $b_j$ of the Table 4.1 by $P(b_j)$. The Table 4.2 is the result of using the product rule to give joint probability distribution for $P(A, B)$.

Table 4.1: An example of $P(A|B)$

| | $b_1$ | $b_2$ | $b_3$ |
|---|---|---|---|
| $a_1$ | 0.4 | 0.3 | 0.6 |
| $a_2$ | 0.6 | 0.7 | 0.4 |

Table 4.2: Calculated joint probability distribution $P(A, B)$

| | $b_1$ | $b_2$ | $b_3$ |
|---|---|---|---|
| $a_1$ | 0.16 | 0.12 | 0.12 |
| $a_2$ | 0.24 | 0.28 | 0.08 |

It should be noted that sum of each column in Table 4.1 is equal to one, whereas the sum of all the entries in Table 4.2 is equal to one. Having estimated the joint probability distribution $P(A, B)$, the probability distribution $P(A)$ can be further calculated. If $a_i$ is the state of $A$ then there are exactly $m$ different events for which $A$ is in state $a_i$, namely mutually exclusive events $(a_i, b_1),\ldots, (a_i, b_m)$. Therefore the following holds:

$$P(a_i) = P(a_i, b_1) + P(a_i, b_2) + \ldots + P(a_i, b_m) = \sum_{j=1}^{m} P(a_i, b_j) \tag{4.10}$$

This calculation (4.10) is called *marginalisation* and we say that variable *B* is marginalised out of *P(A, B)*. The generalised notation for marginalisation is:

$$P(a) = \sum_{B} P(A, B) \tag{4.11}$$

From the Table 4.2, if one marginalises *B*, we get

$$P(a_1)=P(a_1,\ b_1)+P(a_1,\ b_2)+P(a_1,\ b_3)=0.16+0.12+0.12=0.40$$
and
$$P(a_2)=P(a_2, b_1)+P(a_2, b_2)+P(a_2, b_3)=0.24+0.28+0.08=0.60 \tag{4.12}$$

In general, while the product rule is used to construct joint probability distributions, marginalisation reduces a joint probability distribution to a distribution over a subset of its variables.

### *4.2.3 Bayes theorem*

We have so far concentrated largely on the static aspects of the probability theory. But probability is a dynamic theory: it provides a mechanism for coherently revising the probabilities of events as evidence becomes available. Bayesian theorem plays a central role in this. We have introduced the general concept of the Bayesian theorem in Chapter 2 (section 2.8.3) as a learning paradigm and showed that it silently inherits the notions of regularisation and capacity control. The straightforward application of the product rule and the definition of the conditional probability can be used to derive the well-known Bayes theorem. From Eq. (4.3), it follows that $P(A|B)P(B)=P(B|A)P(A)$, which can be written

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{4.13}$$

This general form of the Bayes theorem tells us how to obtain a *posterior* probability on a hypothesis *A* after observation of some evidence *B*, given the *prior* probability in *A* and the *likelihood* of observing *B* where *A* is to be the cause.

In the context of learning models from data, we rewrite the Bayes theorem in the following form:

$$P(H|E,\xi) = \frac{P(H|\xi)P(E|H,\xi)}{P(E|\xi)} \tag{4.15}$$

where we can update our belief in the hypothesis *H* (the model parameters) given the additional evidence *E* (data set of observations) and the background information $\xi$

(background knowledge on the processes which are reflected in our model structure). Using the notation we have introduced in the previous section, the Bayes theorem can be further rewritten as:

$$P(\boldsymbol{\theta}|D,\xi)=\frac{P(\boldsymbol{\theta}|\xi)P(D|\boldsymbol{\theta},\xi)}{P(D|\xi)} \tag{4.16}$$

In order to explain and illustrate all elements in the Eq. (4.16) we will consider the common coin probability problem (Howard, 1970). If we throw the coin up in the air, this event has two possible outcomes (states): a "heads" and a "tails" when the coin lands on the flat surface. Suppose one flips the coin $N$ times and measures the fraction of flips where the outcome is heads. From a classical probability perspective, the long-run fraction of head outcomes is a probability of heads, which is unknown. We can *estimate* this physical probability from the $N$ observations using criteria such as low bias and low variance. Then, we can use this estimate as our probability for heads on the $N$+1 throw. In the Bayesian approach, we also assert that there is some physical probability of heads, but we encode our uncertainty about this physical probability using Bayesian probabilities, and use the Bayes theorem (rule) to compute our probability of heads on the $N$+1 throw.

Let $\Theta$ be the variable whose value $\theta$ may corresponds to the possible true values of the physical probability of heads. The data set of our observations is $D$. We express our uncertainty about $\Theta$ given the background information $\xi$ on that state using the probability density function $p(\theta|\xi)$. Thus, in Bayesian terms, the coin problem reduces to computing $P(x_{N+1}|D,\xi)$ from $P(\theta|\xi)$, that is, computing the posterior probability given the prior probability and the evidence. Using the Eq. (4.16) and having only one parameter $\theta$ we have:

$$P(\theta|D,\xi)=\frac{P(\theta|\xi)P(D|\theta,\xi)}{P(D|\xi)} \tag{4.17}$$

where the normalisation term is

$$P(D|\xi)=\int P(D|\theta,\xi)P(\theta|\xi)d\theta \tag{4.18}$$

Let us consider the term $P(D|\theta,\xi)$. This term is the likelihood function of observing the data, which in this particular case is the binomial distribution function. In particular, given the value of $\Theta$, the observations in $D$ are mutually independent, and the probability of heads (tails) on any observation is $\theta(1-\theta)$. Therefore, Eq. (4.17) becomes:

$$P(\theta|D,\xi)=\frac{P(\theta|\xi)\theta^h(1-\theta)^t}{P(D|\xi)}$$

where $h$ and $t$ the number of heads and tails observed in $D$, respectively. These quantities are sometimes called *hyperparameters* and are said to provide *sufficient statistics* for (in this case) binomial sampling, because they provide sufficient information to compute the

posterior probability from the prior probability. Finally, we average over all possible values of $\Theta$, to determine the probability that the $N$+1 throw of the coin will result as heads:

$$P(X_{N+1}=heads|D,\xi)=\int P(X_{N+1}=heads|\theta,\xi)P(\theta|D,\xi)d\theta$$
$$=\int\theta\, p(\theta|D,\xi)d\theta\equiv E_{P(\theta|D,\xi)}(\theta)$$
(4.20)

where $E_{P(\theta|D,\xi)}(\theta)$ denotes the expectation of $\theta$ with respect to the distribution $P(\theta|D,\xi)$.

Up to this point of discussion, we have considered observations drawn from a binomial distribution. In general, observations may be drawn from any physical probability distribution:

$$P(x|\mathbf{\theta},\xi)=f(x,\mathbf{\theta})$$
(4.21)

where $f(x,\mathbf{\theta})$ is the likelihood function with parameters $\mathbf{\theta}$. For example, $X$ may be a continuous variable and have a Gaussian probability distribution with mean $\mu$ and variance $\sigma$:

$$P(x|\mathbf{\theta},\xi)=(2\pi\sigma)^{-1/2}\, e^{-(x-\mu)/2\sigma}$$
(4.22)

where $\mathbf{\theta}=\{\mu,\ \sigma\}$. However, in most real-world problems the assumption that the observations are drawn from the Gaussian distribution is not valid. In this case a more appropriate prior probability may be a mixture of Gaussian (or other) distributions, e.g.:

$$P(x|\mathbf{\theta},\xi)=0.3\cdot P(x|(\mu_1,\ \sigma_1),\xi)+0.5\cdot P(x|(\mu_2,\ \sigma_2),\xi)+0.2\cdot P(x|(\mu_3,\sigma_3),\xi)$$
(4.23)

In this way, we have introduced an additional *hidden* or unobserved variable $U$, whose states correspond to three possibilities: (i) observations are biased towards low values; (ii) observations are normal and (iii) observations are biased towards high values.

To summarise, regardless of the functional form we can learn about the parameters (model) given data using the Bayes rule. We define the model structure $M$ based on our background knowledge, the we further define variables corresponding to the unknown parameters, assign prior probability to these variables, and use Bayes rule to update our belief about the parameters given the data:

$$P(\mathbf{\theta}|D,M)=\frac{P(\mathbf{\theta}|M)P(D|\mathbf{\theta},D)}{P(D|M)}\qquad posterior=\frac{prior\cdot likelihood}{evidence}$$
(4.24)

We can then average over all possible values of $\mathbf{\Theta}$ to make predictions:

$$P(x_{N+1}|D,\xi\}=\int P(x_{N+1}|\mathbf{\theta},\xi)P(\mathbf{\theta}|D,\xi)d\mathbf{\theta}$$
(4.25)

which integrates out the uncertainty in the model structure and the model parameters. For a class of distribution known as the *exponential* family, these computations can be done efficiently and in a closed form. Members of this class include binomial, multinomial,

normal, Gamma, Poisson, and miltivariate normal distributions (for details see Bernardo and Smith, 1994). In closing this section, we emphasise that, although the Bayesian and classical approaces may sometimes yield same prediction, they are fundamentally different methods for learning from data. The problem of learning the model parameters and the model structure will be further discussed in the context of dynamic Bayesian networks.

## 4.3 Bayesian networks

In this section we introduce some concepts and definition of graph theory that are needed to describe Bayesian networks. Graphs are essential tools for building Bayesian networks and indeed, Bayesian networks are a marriage between probability theory and graph theory.

### *4.3.1 Introduction to graph theory (basic definitions)*

Very many problem domains can be structured through using a graphical representation. Essentially, one identifies the concept or items of information which are *relevant* to the problem under consideration (represented by nodes in the graph), and then makes explicit the *relationships* and *influences* between these concepts.

*Graph:*
A graph $G$ is defined as a pair $G=(V,E)$, where $V=\{V_1, V_2,\ldots, V_n\}$ is finite set of vertices or nodes and $E$ is a subset of the set $V\times V$ called the edges or arcs. Thus, the graph $G$ is simply collection of nodes V and edges $E$ between nodes. These nodes can be connected by edges. If there is a link between two nodes $V_i$ and $V_j$, we use $E_{ij}$ to denote such a link as shown in the following Figure 4.1.

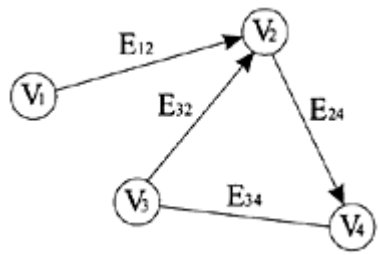



Figure 4.1. Example of graph with directed and undirected links.

The link of a graph can be directed or undirected, depending on whether or not the order of the involved nodes matters.

*Directed link:*

Let *G=(V,E)* be a graph. When $E_{ji} \notin E$, and $E_{ij} \in E$ then the link $E_{ij}$ is called a directed link. A directed link between nodes $V_i$ and $V_j$ is denoted by $V_i \rightarrow V_j$. In Figure 4.1, $E_{12}$ is a directed link as it links $V_1$ and $V_2$ in the direction $V_1$ to $V_2$.

*Undirected link:*

Let $G=(V,E)$ be a graph. when $E_{ij} \in E$ and $E_{ji} \in E$, then the link between nodes $V_i$ and $V_j$ is called an undirected link. An undirected link between nodes $V_i$ and $V_j$ is denoted by $V_i - V_j$ or $V_j - V_i$. In Figure 4.1, $E_{34}$ is an undirected link between the two nodes $V_3$ to $V_4$.

*Parents and children:*

When there is a directed link $V_i \rightarrow V_j$ from $V_i$ to $V_j$, then $V_i$ is said to be a parent of $V_j$ and $V_j$ is said to be a child of $V_i$. In Figure 4.1, $V_1$ is the parent of $V_2$, whereas $V_2$ is the child of $V_1$.

*Directed and undirected graphs:*

A graph in which all the links or edges are directed is called a directed graph. Conversely, a graph in which all the edges are undirected is called an undirected graph. Thus, in a directed graph, the order of the nodes defining a link is important, whereas in an undirected graph, that order is immaterial.

*Cyclic and acyclic graphs:*

A cycle is closed directed path in a directed graph. A directed graph is said to be cyclic if it contains at least one cycle. Otherwise it is called a directed acyclic graph (DAG); see Figure 4.2.

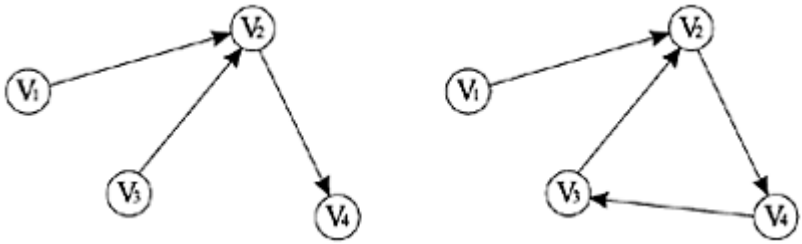



Figure 4.2. Example of an acyclic and a directed cyclic graph.

DAGs play an important role since they are used as a basis for building Bayesian networks. As already mentioned, the important point about a graphical representation of a set of variables is that the edges can be used to indicate relevance or influences between the variables. Absence of an edge between two variables, on the other hand, provides some form of independence statement that nothing about the state of one variable can be inferred by the state of the other. There is a direct relationship between the independence

relationships that can be expressed graphically and the independence relationships that can be defined in terms of probability distributions.

### *4.3.2 Conditional independence*

The notions of independence and conditional independence are fundamental for the probabilistic inference. Detailed studies of the conditional independence properties can be found in Dawid (1979) and Pearl (1988). For completeness, we include definitions and the basic notation after Dawid (1979). The variables *X* and *Y* are *independent* if and only if *P(X,Y)=P(X)P(Y).* The independence is denoted by $X \perp Y$ .If the variables *X* and *Y* are independent then the following can be written:

$$P(X|Y) = \frac{P(X,Y)}{P(Y)} = \frac{P(X)P(X)}{P(Y)} = P(X) \qquad (4.26)$$

Now we introduce a further variable *Z.* Then $X \perp Y|Z$ denotes that *X* is *conditionally independent* of *Y* given *Z.* The following expression shows the conditional independence:

$$P(X,Y|Z)=P(X|Y,Z)P(Y|Z)=P(X|Z)P(Y|Z) \qquad (4.27)$$

As *X* and *Y* are conditionally independent given *Z, Y* drops from the term *P(X|Y,Z)* and results in *P(X|Z).* One can draw a directed acyclic graph that directly encodes this assertion of conditional independence; see Figure 4.3.

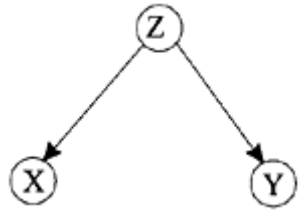



Figure 4.3. Variable *X* is conditionally independent of *Y* given *Z.*

A significant feature of the structure of the graph in Figure 4.3 is that we can now factorise the joint probability distribution of the variables *X, Y* and *Z* into the product of terms that contains at most two variables, thus simplifying the computations:

$$P(X, Y, Z)=P(X, Y|Z)P(Z)=P(X|Z)P(Y|Z)P(Z) \qquad (4.28)$$

As a concrete example, one can think of the variable Z as representing a decrease where the variables *X* and *Y* may represent symptoms. In this configuration, if we observe the decrease at present, the probability of either symptom (*X* or *Y*) being present is

determined. Actual confirmation of one symptom being present will not alter the probability of occurrence of the other symptom.

A different scenario (adopted from Krause, 2000) is illustrated in Figure 4.4, where *X* and *Y* are *marginally independent*, but conditionally dependent given *Z*. For example, both "rain" *(X)* and "sprinkler on" *(Y)* may cause the lawn to become wet. Before any observation of the lawn is made, the probability of rain and the probability of the sprinkler being on are independent. However, once the lawn is observed to be wet, conformation of raining may influence the probability of sprinkler being on (they are alternative causes). This is an example of "explaining away" (Russel and Norvig, 1995), that is, the presence of one cause making an alternative less likely. The probability distribution for this case as shown in Figure 4.4 can be again factorised as:

$$P(X, Y, Z)=P(Z|X, Y)P(X, Y)=P(Z|X, Y)P(X)P(Y) \tag{4.29}$$

It is noted that this is again making use of the independence of *X* and *Y*, $P(X, Y)=P(X)P(Y)$.

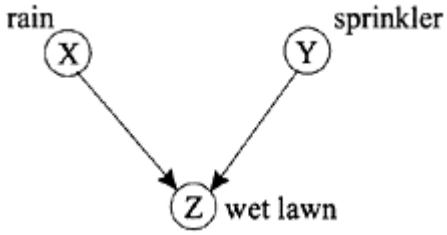



Figure 4.4. Variables *X* and *Y* are conditionally dependent given *Z*.

### *4.3.3 Bayesian networks defined*

A Bayesian network is simply a graphical model that represents conditional independence and efficiently encodes the joint probability distribution (physical or Bayesian) between a set of variables (Heckerman, 1997). In mathematical terms, a Bayesian network for a set of variables $\mathbf{X}=\{X_1, X_2,\ldots, X_n\}$ consists of:

1. A network structure *S* that encodes a set of conditional independence assertions about variables in **X;**
2. A set *P* of local probability distributions associated with each variable. Together, these components define the joint probability distribution for **X**.

The network structure *S* is a directed acyclic graph. The nodes in *S* are in one-to-one correspondence with the variables **X**. We use the notation after Perl (1988), $X_i$ to denote both the variable and its corresponding node, and $\mathbf{Pa}_i$ to denote parents of node $X_i$ in *S*. The lack of possible arcs in *S* encodes conditional independencies. This implies that, given the structure *S*, the Bayesian network provides a complete joint probability distribution for the variables through the equation:

$$P(\mathbf{x}) = \prod_{i=1}^{n} P(x_i | Parents(X_i)) = \prod_{i=1}^{n} P(x_i | \mathbf{pa}_i) \qquad (4.30)$$

The *local probability distributions P* are the distributions corresponding to the terms in the product of Eq. (4.30). Readers familiar with methods for supervised machine learning will recognise that a local distribution function is in fact a probabilistic classification or regression function. Thus, a Bayesian network can be viewed as a collection of probabilistic classification/regression local models organised by conditional-independence relationships. This point of view is in a line with the local modelling in phase space discussed in the previous Chapter 3. The probabilities encoded by a Bayesian network may be Bayesian or physical. When building Bayesian networks from prior knowledge alone, the probabilities will be Bayesian. When learning these networks from data, the probabilities will be physical (and their values uncertain).

In order further to illustrate the process of building Bayesian network we shall consider two examples:

*Example 4.1 (simple weather example)*

Given a situation where it might rain today, and might rain tomorrow, what is the probability that it will rain on both days? Rain on two consequitive days are not independent events with isolated probabilities. If it rains on one day, it is more likely to rain the next day as well. Solving such a problem involves determining the joint probability: chances that it will rain today, and then determining the chance that it will rain tomorrow conditional on the probability that it will rain today. Suppose that P(rain today)=0.20 and *P* (rain tomorrow|it rains today)=0.70. The probability of such joint event is determined by:

$$P(E_1, E_2) = P(E_1)P(E_2|E_1) \qquad (4.31)$$

which also can be expressed as

$$P(E_2|E_1) = \frac{P(E_1, E_2)}{P(E_1)} \qquad (4.32)$$

Working out the joint probabilities, a possible result can be expressed in Table 4.3.

Table 4.3. Marginal and joint probabilities for rain both today and tomorrow

| | Rain tomorrow | No rain tomorrow | Marg. prob. rain today |
|---|---|---|---|
| Rain today | 0.14 | 0.06 | 0.20 |
| No rain today | 0.08 | 0.72 | 0.80 |
| Marg. prob. rain tomorrow | 0.22 | 0.78 | |

From the Table 4.3 it is evident that the joint probability of rain over both days is 0.14., but there is a great deal of other information that had to be brought into the calculations before such a determination was possible. With only two discrete, binary variables, four calculations were required. The same scenario is expressed using a Bayesian network in Figure 4.5 ("!" is used to denote "not")

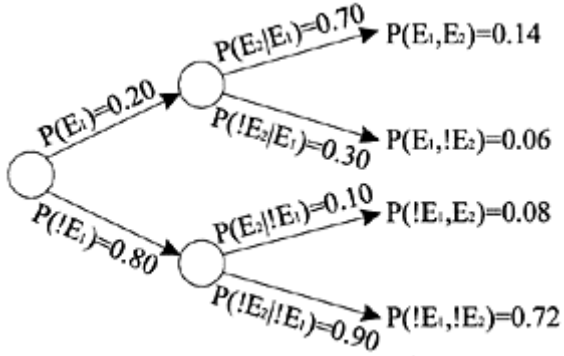



Figure 4.5. An example of Bayesian network showing the probability of rain.

One attraction of the Bayesian network is the efficiency that only one branch of the three needs to be traversed. In this case, we are only concerned with $P(E_1)$, $P(E_2|E_1)$ and $P(E_1, E_2)$. Furthermore, one can also utilise the graph both visually and algorithmically to determine which variables are independent of each other. Instead of calculating four joint probabilities, we can use the independence of the variables to limit our calculations to two. It is self-evident from the Figure 4.5 that the probabilities of rain on the second day having rained on the first are completely autonomous from the probabilities of rain on the second day having not rained on the first.

*Example 4.2 (wet grass)*

This example is adopted from Jensen (1996). Mr. Holmes leaves his house in the morning and notices that his grass is wet. It is due to either rain or has he forgotten to turn off the sprinkler? His belief in both events increases. Next he notices that the grass of his neighbour, Dr. Watson's grass is also wet. Elementary; Holmes is almost certain that it

has been raining, as there is no sprinkler for Watson's grass. A Bayesian network representing the situation described is shown on Figure 4.6.

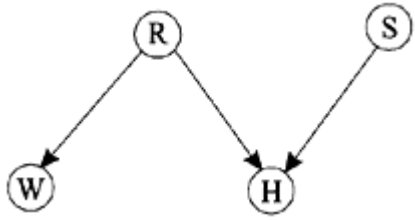



Figure 4.6. A Bayesian network model for the "wet grass" example.

The network consists of four variables namely: Holmes' grass (H), Watsons' grass *(W)*, Rain *(R)* and Sprinkler *(S)*. For sake of simplicity, every variable is assumes to be in only two states "Yes" and "No" denoted by *"y"* and *"n"* respectively. In general, the prior probabilities for rain *(R)* being in state *"y"* and in state *"n"* is denoted $P(R=y)$ and $P(R=n)$. For all others variables also, the same types of notation are used. Let $P(R=y)=0.2$ and $P(R=n)=0.8$, which can be denoted as P(R)=(0.2, 0.8). Similarly, the prior probabilities for sprinkler *S* be *P(S)*=(0.1,0.9). The remaining conditional probabilities are listed in the following Tables 4.4 and 4.5.

Table 4.4 Conditional probabilities for Watsons' grass *(W)*

| Rain*(R)* | Yes *(y)* | No *(n)* |
|---|---|---|
| *W=y* | 1 | 0.2 |
| *W=n* | 0 | 0.8 |

Table 4.5 Conditional probabilities for Holmes' grass (H)

| Rain *(R)* | Yes *(y)* | | No *(n)* | |
|---|---|---|---|---|
| Sprinkler *(S)* | Yes *(y)* | No *(n)* | Yes *(y)* | No *(n)* |
| *H=y* | 1 | 1 | 0.9 | 0 |
| *H=n* | 0 | 0 | 0.1 | 1 |

It should be noted that the Table 4.4 and 4.5 is actually *P(W|R)* and *P(H|R, S)* respectively. In order to calculate the initial probabilities for *W* and *H*, we use the product rule presented in Eq. (4.3) to calculate *P(W, R)* and *P(H, R, S)* as follows:

$P(W=y, R=y)=P(W=y|R=y)P(R=y)=1\cdot 0.2–0.2$
$P(W=y, R=n)=P(W=y|R=n)P(R=n)=0.2\cdot 0.8=0.16$
$P(W=n, R=y)=P(W=n|R=y)P(R=y)=0–0.2–0$
$P(W=n, R=n)=P(W=n|R=n)P(R=n)=0.8–0.8–0.64$

These joint probabilities for *P(W, R)* are tabulated in Table 4.6. This table is result of a multiplication of Table 4.4 by the prior probability of rain *P(R)*.

Table 4.6: Joint probabilities for *P(W, R)*

| Rain *(R)* | Yes *(y)* | No *(n)* |
|---|---|---|
| *W=y* | 0.2 | 0.16 |
| *W=n* | 0 | 0.64 |

From *P(W, R)*, we can marginalise *R* to get the probability distribution *P(W)*. By summing up the rows in Table 4.6 we have:

$P(W=y)=0.2+0.16=0.16$

$P(W=n)=0+0.64=0.64$

The calculation of *P(H, R, S)* follows the same scheme, only the product is:

$P(H, S, R)=P(H|R,S)P(R, S)$.

Since *R* and *S* are independent, we have:

$P(H, S, R)=P(H|R, S)P(R)P(S)$.

The required joint probability distribution for *P(H, R, S)* is obtained by multiplying Table 4.5 by *P(R)P(S)*. For example (see Table 4.7 for complete joint probabilities):

$P(H=y, S=y, R=y)=P(H=y|R=y, S=y)P(R=y)P(S=y)=1\cdot 0.2\cdot 0.1=0.02$

Table 4.7 Joint probabilities for *P(H,R,S)*

| Rain *(R)* | Yes *(y)* | | No *(n)* | |
|---|---|---|---|---|
| Sprinkler *(S)* | Yes *(y)* | No *(n)* | Yes *(y)* | No *(n)* |
| *H=y* | 0.02 | 0.18 | 0.072 | 0 |
| *H=n* | 0 | 0 | 0.008 | 0.72 |

Having computed *P(H, R, S)*, we can marginalise *(R, S)* to get *P(H)*. From Table 4.7 it follows that:

$P(H=y)=0.02+0.18+0.072+0=0.272$

$P(H=n)=0+0+0.008+0.72=0.728$.

### *4.3.4 Inference in Bayesian networks*

The most common task we wish to solve using Bayesian networks is probabilistic inference. Once we have constructed our Bayesian network (model) from prior knowledge, data or a combination of both, we usually need to determine various probabilities of interest from the model. For example, in our wet grass example, we may wish to know the probability of a rain or sprinkler given the observation (evidence) on the Holmes' grass. This probability is not directly stored in the model and needs to be

computed. In general, the computation of a probability of interest given a model and additional evidence is known as *probabilistic inference* (Heckerman, 1997). In other words, inference is the task of efficiently deducing what is the underlying distribution over a particular subset of variables (parameters) given that one knows the states of some other variables in the network. Thus, one needs to calculate a particular conditional or marginal probability distribution.

Inference in Bayesian networks is straightforward when all available evidence is on variables that are ancestor of the variables of interest. However, when evidence is available on a descendant of the variables of interest, one needs to perform inference against the direction of the edges. This can be done by employing the Bayes' theorem. We can illustrate this by using the example 4.2: Let us assume that Watson's grass is wet. Now we are able to determine all the other probabilities given the evidence on *W*. First, the information that Watson's grass is wet is used to update probability of rain *P(R)*. For this, Bayes' theorem is used as:

$$P(R|W=y)=\frac{P(W=y|R)P(R)}{P(W=y)}$$

From Table 4.4, we have $P(W=y|R)$= (1,0.2) and $P(R)$=(0.2,0.8). Thus:

$$P(R|W=y)=\frac{(1,0.2)(0.2,0.8)}{0.36}=(0.56,0.44)$$

This probability is higher than the prior probability of rain $P(R)$=(0. 2,0.8), which is logical because the evidence of Watson's wet grass has increased probability of rain. Now to update probability of *H*, we use fundamental product rule as follows:

$P(H, R, S)=P(H|R, S)P(R)P(S)$

Similarly, joint probability distribution for $P(H, R, S)$ is obtained by multiplying Table 4.5 by $P(R)P(S)$. The following Table 4.8 shows the updated joint probabilities.

Table 4.8 Updated joint probabilities for $P(H,R,S)$

| Rain *(R)* | Yes *(y)* | | No *(n)* | |
|---|---|---|---|---|
| Sprinkler *(S)* | Yes *(y)* | No *(n)* | Yes *(y)* | No *(n)* |
| *H=y* | 0.056 | 0.504 | 0.0396 | 0 |
| *H=n* | 0 | 0 | 0.0044 | 0.396 |

Finally, *P(H)* can be computed by marginalising *(R, S)* from Table 4.8, which results in $P(H)$=(0.5996, 0.4004).

For problems with many variables, the above direct approach is not practical. However, it is possible to evaluate the marginal probability of all variables given observations on some other variables. The problem of conditioning Bayesian networks on observations is in general NP-hard (Cooper, 1990), but experience shows that in many real systems the networks are sparsely connected and therefore the calculations are tractable. Several researchers have developed probabilistic inference algorithms for

Bayesian networks with discrete variables that exploit conditional independence roughly. One of the first algorithms in singly connected networks was developed by Pearl (1988). He developed a message passing scheme, based on the d-separation principle, in which each time a variable received some new evidence it sent a message to its neighbours, which sent new messages to their neighbours. This process is called the local message passing process. Local computation in a Bayesian network is the process of computing a variable's posterior probability distribution from the posterior distribution of its neighbours- and only its neighbours (Mayo, 2001). Thus, when evidence arrives at a node, its neighbours update themselves, then their neighbours update themselves, and so on, until the entire network absorbs the evidence. Inference via local computation is highly efficient for singly connected Bayesian networks.

However, the situation is more complex when the network is multiply connected. Later on, Lauritzen and Spiegelhalter (1988), Jensen et al. (1990) created an algorithm for multiply connected networks that first transforms the Bayesian networks into a tree where each node in the tree corresponds to a subset of variables. The algorithm then exploits several mathematical properties of this tree to perform probabilistic inference. The most popular algorithm used for discrete variables today is the junction tree algorithm designed by the Odin group at Aarhus University (Jensen, 1996).

Methods for exact inference in Bayesian networks with continuous variables that encode multivariate-Gaussian or Gaussian-mixture distributions have been developed by Shacher and Kenely (1989) and Lauritzen (1992), respectively. These methods also use assertions of conditional independence to simplify the inference. Approximate methods for inference in Bayesian networks with other distributions have also been developed (Saul et al., 1996; Jaakkola and Jordan, 1996), which utilise Monte-Carlo methods. When a Bayesian network structure contains undirected cycles, inference in principle is intractable. However, for many practical applications, where the generic inference methods are impractical, researchers are developing techniques that are custom tailored to particular network topologies (Heckerman, 1989; Shacher et al, 1990; Jensen and Andersen, 1990; Darwiche and Provan 1995).

Before closing this section on the "static" Bayesian networks, that is, they do not deal with temporal (time-series) data, we would like to mention some of the practical successful applications of Bayesian networks and the ongoing research in this area.

*AutoClass project*

The National Aeronautic and Space Administration (NASA) has a large investment in Bayesian research. In gathering data from the deep-space observatories and planetary probes, an *apriori* imposition of structure or patterns expectations is inappropriate. The AutoClass project was aimed to develop Bayesian applications that can automatically perform pattern recognition and classification of the raw data. An applied example of AutoClass's application was the input of infrared spectra. Although no differences among this spectra were initially sustected, AutoClass sucesfully distinguished two subgroups of stars (see Cheeseman and Stutz, 1995 for details). Velickov and Solomatine (2000) successfully applied the AutoClass system for unsupervised classification of surge data along the Dutch coast.

*Lumiere project*

Microsoft began work in 1993 on Lumiere, its project to create software that could automatically and intelligently interact with users by anticipating the goals and needs of these users (Horvitz, 1998). This research started as a continuation on earlier research on pilot-aircraft interaction and resulted in the "Office Assistant" product with the introduction of the Office suite of desktop applications, and is nowadays probably the most-sophisticated personal agent based on Bayesian inference.

*Autonomy project*

Autonomy project started as European research project in 1991. Its goal was to create text mining and information retrieval engine that will infer information goals from free-text queries, based on Bayesian inference. This project resulted in a knowledge management suite of tools fully based on Bayesian networks.

In the context of this work, Shrestha (2002) has applied Bayesian networks for risk assessment of structural collapse of sewer, which resulted in very encouraging performance of the proposed Bayesian network.

### 4.4 Dynamic Bayesian networks

Up to this point of discussion, we have introduced only static Bayesian networks. These kinds of networks are useful for solving diagnostic type of problems, since they cannot deal with temporal data. However, learning models from data that describe dynamical systems, which is the focus of this thesis, require incorporation of the temporal order of the processes and variables in the network. This concept plays an important role in the design of *dynamic Bayesian networks*. In time series modelling, we observe the values of certain variables at different points in the time. The assumption that an event can cause another event in the future, but not vice-verse, simplifies the design of Bayesian networks for time series. That means directed arcs should flow forward in time. Assuming a time index t to each variable, one of the simplest causal models for a time series data $\{Y_1,Y_2,\ldots,Y_T\}$ is a *first-order Markov model*, in which each variable is directly influenced only by the previous variable; see Figure 4.7.

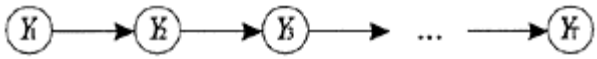



Figure 4.7. A Bayesian network representing a first-order Markov process.

The joint probability distribution of a sequence of observation $\{Y_1, Y_2,\ldots,Y_T\}$ can always be factorised using first-order Markov model as:

$$P(Y_1, Y_2,\ldots, Y_T)=P(Y_T|Y_1, Y_2,\ldots,Y_{T-1})\cdot P(Y_1, Y_2,\ldots, Y_{T-1})$$
$$=P(Y_T|Y_{T-1})\cdot P(Y_1, Y_2,\ldots, Y_{T-1})$$
$$=P(Y_T|Y_{T-1})\cdot P(Y_{T-1}|Y_1, Y_2,\ldots, Y_{T-2})\cdot P(Y_1, Y_2,\ldots, Y_{T-2})$$
$$=P(Y_T|Y_{T-1})\cdot P(Y_{T-1}|Y_{T-2})\ldots P(Y_2|Y_1)\cdot P(Y_1)$$

This form of Eq. (4.13) is called Markov chain rule of first order. Using the shorthand notation $Y_1^T$ to denote sequences from $t$=1,…, $T$ the above equation can be written in compact form as:

$$P\left(Y_1^T\right) = P(Y_1)\prod_{t=2}^{T} P\left(Y_t \middle| Y_{t-1}\right) \tag{4.34}$$

Thus, dynamic Bayesian networks are a special case of singly connected Bayesian networks specifically aimed at time series modelling. In this case, one assumes causal dependencies between events in time leading to a simplified network structure, such as the one shown in Figure 4.7. Namely, in its simplest form, the states of some dynamical system described as a dynamic Bayesian networks satisfy the following first order Markovian condition (Pavlovic, 1999): the state of a system at time $t$ depends only on its immediate past: its state at time $t-1$. These models do not directly represent dependencies between observable over more than one time step. Having observed $\{Y_1, Y_2,\ldots, Y_T\}$, the model will only make use of $Y_T$ to predict the value of $Y_{T+1}$. One simple way of extending Markov models is to allow higher order interactions between variables. For example, a $\tau^{th}$ order Markov model allows arcs from $\{Y_{t-\tau},\ldots, Y_{t-1}\}$ to $Y_t$. The following Figure 4.8 shows the Markov model of order 2, where conditional probability $P(Y_T|Y_1, Y_2,\ldots, Y_{T-1})$ can be simply replaced by $P(Y_T|Y_{T-1}, Y_{T-2})$.

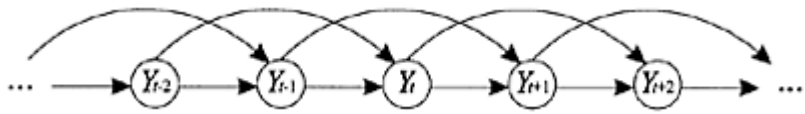



Figure 4.8. A Bayesian network representing a second order Markov process.

Thus, in general, a Markov model of order $k$ is a probability distribution over a sequence of variables $\{Y_1, Y_2,\ldots, Y_T\}$ with the following conditional independence property:

$$P(Y_t|Y_1,\ldots,Y_{t-1}) = P(Y_t|Y_{t-k},\ldots, Y_{t-1}) \tag{4.35}$$

Since $\{Y_{t-k},\ldots, Y_{t-1}\}$ summarises all the relevant past information, $Y_t$ is generally called a *state variable*. Because of the above conditional independence property, the joint distribution of a whole sequence can be decomposed into the product as:

$$\left(Y_1^T\right) = P(Y_1,\ldots,Y_k)\prod_{t=k+1}^{T} P\left(Y_t \middle| Y_{t-k},\ldots,Y_{t-1}\right). \tag{4.36}$$

This equation is the generalised form of the Eq. (4.34). The Markov model of order 1 is completely specified by the so-called *initial state probabilities* $P(Y_1)$ and *transition*

*probabilities* $P(Y_T|Y_{T-1})$. Before we go further in the generalisation of the dynamic Bayesian networks, let us illustrate the first order Markov model using the following example (adopted from Lussier, 1998).

*Example 4.3 (weather types)*

Let us consider three types of weather: sunny *(S)*, rainy *(R)*, and foggy *(F)* and assume for the moment that the weather lasts all day i.e. it does not change from rainy to sunny in the middle of the day. Let us try to predict weather based on the previous history of observations of weather. The simplified model of weather prediction will be: we will collect statistics on what the weather will like today based on what the weather was like yesterday, the day before, and so forth. However, if assume first order Markov model, the tomorrow's weather will depend on only today's weather. The following table shows probabilities of tomorrow's weather based on today's weather $P(W_{tomorrow}|W_{today})$.

Table 4.9. Probabilities for $P(W_{tomorrow}|W_{today})$.

| | | Tomorrow's weather | |
|---|---|---|---|
| Today's weather | Sunny *(S)* | Rainy *(R)* | Foggy *(F)* |
| Sunny *(S)* | 0.8 | 0.05 | 0.15 |
| Rainy *(R)* | 0.2 | 0.6 | 0.2 |
| Foggy *(F)* | 0.2 | 0.3 | 0.5 |

For first-order Markov models, we can use these probabilities to draw a probabilistic finite state automaton. For the weather domain, we would have three states (Sunny *(S)*, Rainy *(R)*, Foggy *(F)*), and every day we would transition to a possibly new state based on the probabilities in Table 4.9. Such an automaton is schematically presented in Figure 4.9.

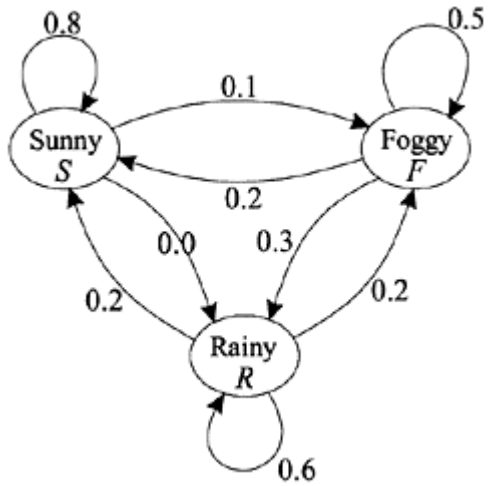


Figure 4.9. Schematisation of the simple weather model.

This simple model allows to make various probabilistic inferences such as:

• Given that today is sunny, what is the probability that tomorrow is sunny and the day after is rainy?

Applying the product rule and then the first-order Markov assumption, we have:

$$P(W_2=S,W_3=R|W_1=S)=P(W_3=R|W_1=S,W_2=S)\cdot P(W_2=S|W_1=S)$$
$$=P(W_3=R|W_2=S)\cdot P(W_2=S|W_1=S)$$
$$= 0.05\cdot 0.8=0.04$$

• Given that today is foggy, what is the probability that it will be rainy two days from now?

There are three possible sequences to get from foggy today to rainy two days from now: *{F, F, R}, {F, R, R}, {F, S, R}.* Therefore one has to sum over these paths:

$P(W_3=R|W_1=F)=P(W_2=F,W_3=R|W_1=F)+P(W_2=R,W_3=R|W_1=F)+ P(W_2=S, W_3=R|W_1=F)$

$= P(W_3=R|W_1=F,W_2=F)P(W_2=F|W_1=F)+ P(W_3=R|W_1=F,W_2=R)P(W_2=R|=F)+ P(W_3=R|W_1=F,W_2=S)P(W_2=S|W_1=F)$

$=P(W_3=R|W_2=F)P(W_2=F|W_1=F)+ P(W_3=R|W_2=R)P(W_2=R|W_1=F)+ P(W_3=R|W_2=S)P(W_2=S|W_1=F)$

$=0.3\cdot 0.5+ 0.6\cdot 0.3+ 0.05\cdot 0.2=0.34$

### *4.4.1 State-space models*

As already mentioned, these simple Markov models can be extended to include dependencies between the observables using higher-order Markov process. However in perspective of dynamic Bayesian networks, another way to extend Markov model is to conceive a higher conceptualisation level that the observations are dependent on a *hidden variable,* which we call state, and that the *sequence of states* is modelled as a Markov process; see Figure 4.10.

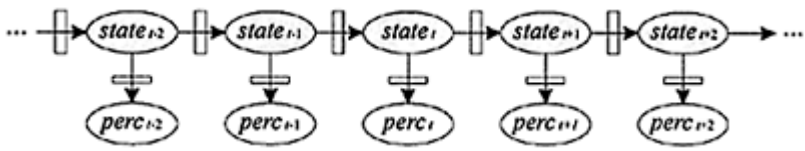


Figure 4.10. Generic structure of a dynamic Bayesian network.

A generic dynamic Bayesian network is structured as a sequence of time slices, where the nodes at each slice encode the state at the corresponding time point. The conditional probability distributions encode both a *state evolution model*, which describe the transitional probabilities between the states, and a *sensor model*, which describes the observations that can result from a given state. Typically, one assumes that these distributions do not vary over time, although this is also possible, as we will discuss in Chapter 5. A classic model of this kind is the linear-Gaussian state-space model, also known as Kalman filer, see Figure 4.11.

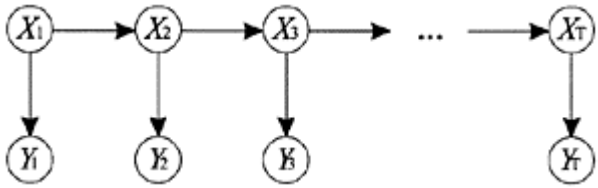



Figure 4.11. A dynamic Bayesian network specifying conditional independence relation for a state-space model.

In state-space models, a sequence of *D*-dimensional real-valued observation vectors $\{Y_1, Y_2,\ldots, Y_T\}$, is modelled by assuming that at each time step $Y_t$ was generated from a *K*-dimensional real-valued hidden state variable $X_t$, and that the state evolution model is defined by first-order Markov process. The joint probability of the state-space model can be expressed as:

$$P(\{X_t, Y_t\}) = P(X_1)P(Y_1|X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})P(Y_t|X_t). \qquad (4.37)$$

The state transition probability $P(X_t|X_{t-1})$ is decomposed into deterministic and stochastic components:

$$X_t=f_t(X_{t-1})+\omega_t \qquad (4.38)$$

where $f_t$ is the deterministic transition function determining the mean of the $X_t$ given $X_{t-1}$, and $w_t$ is a zero-mean random noise vector. Similarly, the emission model $P(Y_t|X_t)$ is decomposed as:

$$Y_t=g_t(X_t)+v_t. \qquad (4.39)$$

If both the transition and output function are linear and time-invariant and the distribution of the states and observation noise variables is Gaussian, the model becomes a linear-Gaussian state-space model:

$$X_t = AX_t\,1 + \omega_t$$
$$Yt = CXt + v_t \qquad (4.40)$$

where $A$ is the state transition matrix and $C$ is the observation matrix. Linear Gaussian state-space models (sometimes referred as linear dynamic models) are used extensively in all areas of control and signal processing. In the area of data-driven modelling they have recently showed promising results in the area of data assimilation for physically-based models (e.g. Babovic, 2000).

### *4.4.2 Hidden Markov models*

It would be intractable in general to model sequential data in which the conditional probability distribution $P(Y_T|Y_1, Y_2,\ldots, Y_{T-1})$ of an observed variable $Y_T$ at time $t$ depends on all the details of the previous values $Y_1, Y_2,\ldots, Y_{T-1}$ (Bengio, 1999). Even with Markov models of order $k$ the problem is that they quickly become intractable for large $k$. For example, for a multinomial state variable $Y_t \in \{1,\ldots,n\}$, the number of required parameters for representing the transition probabilities is of order $O(n^{k+1})$. This necessarily restricts one to using a small value of $k$.

However it is possible to model sequential data using the concept of hidden variables which can summarise a past sequence concisely. In other words, the hidden variables which are *unobserved variables* carry all the information from $Y_1, Y_2,\ldots, Y_{T-1}$ that is useful to describe the distribution of the next observation $Y_T$. This is precisely what Hidden Markov Models (HMMs) embed: we do not assume that the observed data sequence has a Markov property of low order; however, another, unobserved but related variable (the state variable) is assumed to exist and to have the Markov property (with low order, typically $k$ =1). HMMs are generally taken order of 1 because a HMM of order 1 can emulate an HMM of any order by increasing the number of values that the state variable can take (for details see MacDonald and Zucchini, 1997).

In mathematical terms, the joint probability for the sequences of hidden states $\{S_t\}$ and observations $\{Y_t\}$ can be factored in exactly the same manner as Eq. (4.37), with $S_t$ taking place of $X_t$:

$$P(\{S_t, Y_t\}) = P(S_1)P(Y_1|S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_t|S_t). \qquad (4.41)$$

Consequently, the conditional independences in an HMM can be also expressed graphically using dynamic Bayesian network shown in Figure 4.12.

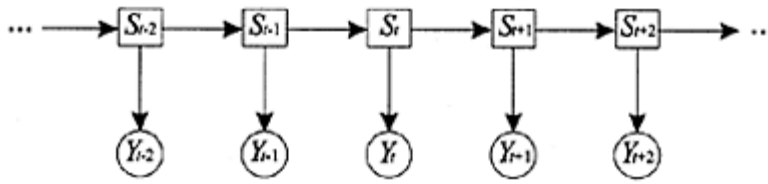


Figure 4.12. A dynamic Bayesian network specifying conditional independence assumption for a Hidden Markov Model of first order. Square is used to represent the discrete hidden variable.

The joint distribution (Eq. 4.41) is therefore completely specified in terms of:

(i) The initial state probabilities $P(S_t)$
(ii) The transition probabilities $P(S_t|S_{t-1})$
(iii) The emission probabilities $P(Y_t|S_t)$

All of the conditional probability distribution can be time-varying $P(S_t|S_{t-1})=(S_t|S_{t-1}, t)$ or time invariant, parametric $P(S_t|S_{t-1})=(S_t|S_{t-1}, \theta)$ or non parametric (conditional probability tables). Depending on the type of the state space of hidden and observable variables, a HMM can be discrete, continuos, or a combination of two. In general, the hidden state is represented by a single multinomial variable that can take one of $K$ discrete values, $S_t \in \{1,...,K\}$. Thus, state transition probabilities, $P(S_t|S_{t-1})$ for time-invariant HMM can be specified by a single $K \times K$ transition matrix. If the observed variables are discrete symbols taking on one of $L$ values, the emission probability $P(Y_t|S_t)$ can be fully specified by a $K \times L$ observation matrix. For real value observation vectors, $P(Y_t|S_t)$ can be modelled in many different forms, such as a Gaussian, mixtures of Gaussian or a neural network. HMMs can be also extended to allow for input variables, known as input-output HMMs (e.g. Bengio and Frasconi, 1995). The system then models the conditional distribution of a sequence of output observations given a sequence of input observations. These kinds of HMMs have been extensively applied to problems of bioinformatics (Krogh et al., 1994; Baldi et al., 1994), speech recognition and (Juang and Rabinner, 1991) and system identification (Smyth, 1994).

In order to illustrate the concept of HMMs, let us consider an example (adopted from Wolfgong, 1999) where the HMM is a four-state model as shown in the following Figure 4.13. The state sequence $\{St\}=(S_1, S_1, S_2, S_3, S_3, S_4)$ generates the observation sequence $\{Y_t\}=(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6)$. When progressing through the model, at each time $t$ that a state $S_j$ is entered, an observation $Y_m$ is generated with probability $b_{mj}$. The transition from state $i$ at time $t-1$ to state $j$ at $t$ time is governed by the discrete state bigram transition probability $a_{ij}$. The observation probabilities attain their maxima when corresponding state is visited. Therefore in Figure 4.13, the maximum values for $b_{mj}$ are obtained for $b_{11}$, $b_{21}$, $b_{32}$, $b_{43}$, $b_{53}$, and $b_{64}$.

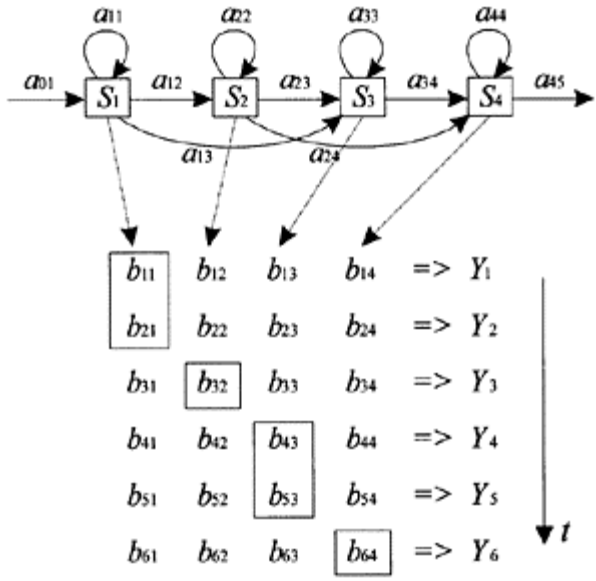



Figure 4.13. An example of a four-state HMM model.

The probability that the observation sequence $\{Yt\}=(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6)$ is generated given the model and moving through the state sequence $\{S_t\}=(S_1, S_1, S_2, S_3, S_3, S_4)$ is calculated simply as a product of the state transition and corresponding maximum observation probabilities as:

$$P(\{St,Yt\})=a_{01}b_{11}a_{11}b_{21}a_{12}b_{32}a_{23}b_{43}a_{33}b_{53}a_{34}b_{64} \tag{4.42}$$

In practice, only the observation sequence $\{Y_t\}$ is known and the underlying state sequence $\{S_t\}$ is hidden as explained above. Given that $\{S_t\}$ is unknown, the required likelihood is computed by summing up the probabilities over all possible state sequences, given the observation sequence. More precisely:

$$P(Y)=\sum_{S} P(S_{t+1}|S_t)\prod_{t=1}^{T} P(Y_t|S_t)P(S_t|S_{t-1}) \tag{4.43}$$

where $S_o$ is constrained to the model entry state, and $S_{t+1}$ to be the model exit state. As an alternative to the Eq. (4.43), the likelihood can be approximated by only considering the most likely state sequence, i.e.

$$P(Y)=\max_{S}\left\{ P(S_{t+1}|S_t)\prod_{t=1}^{T} P(Y_t|S_t)P(S_t|S_{t-1})\right\} \tag{4.44}$$

The direct computation of Eq. (4.43) or Eq. (4.44) is intractable. Simple recursive procedures are used to calculate these quantities very efficiently, assuming that

$$P(Y|S)=P(Y|\theta) \tag{4.45}$$

and the model parameters θ are known. Given a set of training examples and a particular model, the parameters of that model can be determined by a robust and efficient reestimation procedure (to be further discussed).

### *4.4.3 Switching state-space models*

In order to model a time series with continuous but non-linear dynamics, it is possible to combine the real-valued hidden state of linear-Gaussian state-space models and the discrete state of HMMs (Ghahramani and Hinton, 1998). One natural way to do this is the switching state-space models. In switching state-space models, the sequence of observations $\{Y_t\}$ is modelled using a hidden state space comprising of M real-valued state space vectors, $\left\{X_t^{(m)}\right\}$, and one discrete state vector $\{S_t\}$. The discrete state is a multinomial variable that can take on $M$ values: $S_t \in \{1,\ldots M\}$ and sometimes is referred to as a switch variable. The joint probability of observations and hidden states can be factored as:

$$P\left(\left\{S_t, X_t^{(1)},\ldots, X_t^{(M)}, Y_t\right\}\right)= P(S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})\prod_{m=1}^{M} P\left(X_1^{(m)}\right)\prod_{t=2}^{T} P\left(X_t^{(m)}\middle|X_{t-1}^{(m)}\right) \times\prod_{t=1}^{T} P\left(Y_t|X_t^{(1)},\ldots, X_t^{(M)}, S_t\right) \tag{4.46}$$

which corresponds graphically to the conditional independences represented in Figure 4.14. Conditioned on a setting of the switch variable $S_t=m$, the observed variable is multivariate Gaussian with output defined by the state-space model $m$. The probability of the observation vector $\{Y_t\}$ can be expressed as:

$$P\left(Y_t|X_t^{(1)},\ldots, X_t^{(M)}, S_t = m\right)=(2\pi)^{-D/2}|R|^{-1/2}\times \exp\left\{-\frac{1}{2}\left(Y_t - C^{(m)}X_t^{(m)}\right)^{\prime} R^{-1}\left(Y_t - C^{(m)}X_t^{(m)}\right)\right\} \tag{4.47}$$

where $D$ is the dimension of the observation vector, $R$ is the observation noise covariance matrix, and $C^{(m)}$ is the output matrix for state-space model (recall Eq. 4.40).

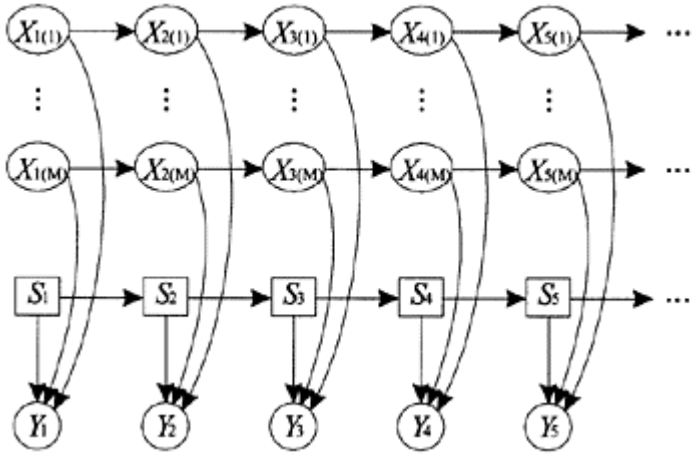



Figure 4.14. Dynamic Bayesian network representation for switching state-space models. $S_t$ is the discrete switch variable and $X_t^{(m)}$ are the real-valued state vectors.

Each real-valued state space vector evolves according to the linear-Gaussian dynamics of a state-space model with differing initial state, transition matrix, and state noise. The switch state itself evolves according to the discrete Markov dynamics specified by initial state probabilities $P(S_1)$ and an $M \times M$ state transition matrix $P(S_t|S_{t-1})$.

### 4.5 Learning dynamic Bayesian networks

How can dynamic Bayesian networks (DBNs) learn from data? There are several variants of this question. The structure of the DBN can be *known* or *unknown*, and the variables in the network describing the dynamics of the system can be *observable* or *hidden* in all or some of the data points. Learning the structure (and the parameters at the same time) of a DBN is a difficult problem and is closely related to machine learning. As we have already argued in Chapter 2, our approach of learning models from data is based on a maximum utilisation of the expert knowledge about the physical processes and the relationships between variables characterising those processes that are encapsulated in the model structure and model parameters. Thus, our assumption of learning DBNs is that the learning process starts with some *a priori* knowledge about the model(s) structure (the architecture of the DBN) and the model parameters. This initial knowledge is represented in the from of *a priori* probability distribution over possible model structures and parameters, and updated using the time series of observables to obtain a posterior probability distribution over the models and parameters. In Section 4.2.3 we have already described the process of learning posterior probabilities using the Bayesian framework.

More formally, assuming a prior distribution over models structures *P(M)* and a prior distribution over parameters for each model structure $P(\theta|M)$, a data set of time series of observables *D* is used to form a posterior distribution over models using Bayes rule:

$$P(M|D) = \frac{P(M)\int P(D|\theta, M)P(\theta|M)d\theta}{P(D)} \quad . \tag{4.48}$$

For a given model structure, one can compute the posterior distribution over the parameters:

$$P(\theta|D, M) = \frac{P(\theta|M)P(D|\theta, M)}{P(D|M)} \quad . \tag{449}$$

Given the data set of observations $D=\{Y_1, Y_2,\ldots,Y_T\}$, one can predict the next observation $Y_{T+1}$ using the Bayesian framework as:

$$P(Y_{T+1}|D)=\int P(Y_{T+1}|\theta,D, M)P(\theta|D, M)P(M|D)d\theta dM \tag{4.50}$$

which integrates out the uncertainty in the model structure and parameters. Using this learning approach we obtain the somewhat limited case of the Bayesian approach to learning if we assume a single model structure M and we estimate the model parameters $\hat{\theta}$ that maximises the likelihood of observations $P(D|\theta, M)$ given that model. Since we focus in this work on estimating model parameters given the model structure(s), in principle this is an only approximate Bayesian learning, due to the reason that in practical applications a full-fledged Bayesian analysis is often impractical. Some researchers have developed approximate methods for integrating over the posterior distribution in the case of neural network models (e.g. Neal, 1996; MacKay, 1995).

### *4.5.1 Learning with complete data*

The most straightforward learning situation to consider is that where the structure of the DBN is (believed to be) known and data is observable on all variables in the network. Assuming a data set of observables $D=\{y^{(1)}, Y^{(2)},\ldots, Y^{(N)}\}$, each of which can be a time series of vectors, then the likelihood of the data set is:

$$P(D|\theta, M) = \prod_{i=1}^{N} P(Y^{(i)}|\theta, M). \tag{4.51}$$

Since we assume that the model structure is known, from this point of the discussion we can drop the implicit conditioning on the model structure, *M*. The model parameters can be obtained by maximising the likelihood, or equivalently the *log* likelihood:

$$L(\theta) = \sum_{i=1}^{N} log P\left(Y^{(i)} \middle| \theta\right). \tag{4.52}$$

Since the observation vector includes all variables in the Bayesian network, each term in the log likelihood can be factored as:

$$log\ P\left(Y^{(i)} \middle| \theta\right) = log \prod_{j} P\left(Y_j^{(i)} \middle| Y_{pa(j)}^{(i)}, \theta_j\right) \tag{4.53}$$

where *j* indexes the nodes in the network, *pa(j)* is the set of parents of j, and $\theta_j$ are the parameters that define the transitional probabilities. The likelihood therefore decouples into local terms involving each node and its parent(s), thus simplifying the estimation problem. Any optimisation algorithm (e.g. gradient-based, see Section 2.9) can be used to maximise the log likelihood given with Eq. (4.52).

### *4.5.3 Learning with incomplete data*

Let us now discuss methods for learning about parameters when the data set is incomplete (i.e. some variables in some cases are not observed or are hidden). An important distinction concerning missing data is whether the absence of an observation is dependent on the actual states of the variables describing the dynamical system. For example, a missing datum in a study of surge dynamics along the coast may indicate the presence of a stormy situation. In contrast, if a variable is hidden (i.e. is not observable in any case), then the absence of the data is independent of the states. Although Bayesian methods are suited to the analysis of both situations, methods for handling missing data where absence is independent of states (i.e. presence of hidden variables) are simpler than those where the absence and states are dependent. In this work we concentrate on the problem of learning DBNs with hidden variables. We refer readers interested in the more complicated case to Pearl (1995). With hidden variables, the exact computation of posterior distribution of the model parameters is intractable, meaning that the log likelihood cannot be decomposed as in Eq. (4.53). In this case we have:

$$L(\theta) = log\ P(Y|\theta) = log \sum_{X} log\ P(Y, X|\theta) \tag{4.54}$$

where *X* is the set of hidden variables, and $\sum_X$ is the sum (or integral) over *X* required to obtain the marginal probability of the data. Thus, one requires approximate methods.

A number of such approximations have been described in the literature. One can broadly divide them into two classes: stochastic and deterministic. The Stochastic class of approximations is based on Monte-Carlo or sampling methods. These approximations can be extremely accurate, but computationally demanding. A widely studied stochastic method is *Gibbs sampling* (Geman and Geman, 1984), which is a special case of the general Markov chain Monte-Carlo methods for approximate inference. Given variables $\{X_1, X_2, \ldots, X_n\}$ with some joint distribution *p(x)*, we can use Gibbs sampling to approximate the expectation of a function *f(x)* with respect to *p(x)* as follows. First, we

assign an initial state for each variable in *{X}*. Next, we pick some variable $X_i$, unassign its current state, and compute its probability distribution given the state of the other n-1 variables. Then, we sample a state for $X_i$ based on its probability distribution, and compute *f(x)*. Finally, we iterate the previous two steps, keeping track of the average value of *f(x)*. In the limit, as the number of cases approaches infinity, this average is equal to the mathematical expectation $E_{p(x)}(f(x))$, provided that two conditions are met. First, the Gibbs sampler must be *irreducible*, and second each variable $X_i$ must be chosen infinitely (see Neal, 1993 for discussion).

Monte-Carlo methods yield accurate results, but they are often impractical for large data sets. Another approximation that leads to accurate results for relatively large data samples is the *Gaussian approximation* (Kass and Raftery, 1995). The idea behind this is that, for large amounts of data, $P(\theta|D,M) \propto P(D|\theta,M)\cdot P(\theta|M)$ can be approximated with a multivariate-Gaussian distribution. In particular, if we denote:

$$g(\theta) \equiv log(P(D|\theta, M)\cdot P(\theta|M)) \tag{4.55}$$

and let $\hat{\theta}$ be the estimation of θ that maximises $g(\theta)$. Then this configuration also maximises $P(\theta|D, M)$, and is known as the *maximum a posteriori* (MAP) configuration of the model parameters θ. As the sample size of the data increases, the Gaussian peak will become sharper, tending to a delta function at the MAP estimation $\hat{\theta}$. In this limit, one does not need to compute averages or expectations. Instead, one simply makes predictions based on the MAP configuration.

A further approximation is based on the observation that, as the sample size increases, the effect of the prior $P(\theta|M)$ diminishes. Thus, one can approximate $\hat{\theta}$ by the *maximum likelihood* (ML) configuration of model parameters θ:

$$\hat{\theta} = \arg\max_{\theta} \{P(D|\theta, M)\} . \tag{4.56}$$

As already previously discussed, one class of maximisation techniques for finding MAP or ML is gradient-based optimisation. For example, using gradient ascent one can follow the derivatives of $g(\theta)$ or the likelihood $P(D|\theta, M)$ to a local maximum (e.g. Russel et al., 1995). Of course, these gradient-based methods find only local maxima.

Another alternative technique for finding a local ML or MAP is the *expectation-maximisation* (EM) algorithm, introduced by Dempster et al., (1977). This algorithm can be viewed as a deterministic version of the Gibbs sampling. The EM algorithm iterates through two steps: expectation—E step, and maximisation—M step. We shall here describe the main concept of the EM algorithm, since a more detailed explanation is given in Chapter 5. In the context of learning DBNs with hidden variables, the log likelihood is expressed as Eq. (4.54). Using any distribution *Q* over the hidden variables, one can obtain a lower bound on $L(\theta)$ as:

$$\log \sum_X P(Y,X|\theta) = \log \sum_X Q(X) \frac{P(Y,X|\theta)}{Q(X)} \geq \sum_X Q(X) \log \frac{P(Y,X|\theta)}{Q(X)} \tag{4.57}$$
$$= \sum_X Q(X) \log P(Y,X|\theta) - \sum_X Q(X) \log Q(X) = F(Q,\theta)$$

where the inequality is known as Jensen's inequality and can be proven using the concavity of the log function (see e.g. Neal and Hinton, 1993). If one defines the energy of the global configuration *(X, Y)* to be *log P*(*Y, X*|θ), then the lower bound $F(Q, \theta) \leq L(\theta)$ is the negative of the quantity known in statistical physics as *the free energy:* the expected energy under *Q* minus the entropy of *Q* (Hinton and Zemel, 1994). Thus, the EM algorithm alternates between maximising *F* with respect to *Q* and θ, respectively, holding the other fixed. It is thus coordinate ascent in *F*. Starting with some initial parameters $\theta_0$:

E step:
$$Q_{k+1} \leftarrow \arg\max_Q F(Q, \theta_k) \tag{4.58}$$

M step:
$$\theta_{k+1} \leftarrow \arg\max_\theta F(Q_{k+1}, \theta) \,. \tag{4.59}$$

The EM algorithm is fast, but it has the disadvantage of not providing a distribution over the model parameters θ. In addition, Lauritzen (1995) reported that when a substantial amount of data is missing, the likelihood function has a number of local maxima leading to poor results (which is also problem with gradient-based methods). Several improvements have been suggested to overcome this problem (e.g. Buntine, 1995).

In the light of the previous introduction to the dynamic Bayesian networks, state-space models and HMMs in particular, we present the learning of the model parameters with the presence of hidden variables at the end of this section.

## LEARNING STATE-SPACE MODELS

Recalling Eq. (4.37), the *log* probability of the hidden states and the observations for linear-Gaussian state-space models can be written as:

$$\log P(\{X,Y\}) = \log P(X_1) + \sum_{t=1}^{T} \log P(Y_t|X_t) + \sum_{t=2}^{T} \log P(X_t|X_{t-1}). \tag{4.60}$$

In Eq. (4.60) each of the conditional probabilities terms is Gaussian, e.g. using Eq. (4.40):

$$\log P(Y_t|X_t) = -\frac{1}{2}(Y_t - CX_t)' R^{-1}(Y_t - CX_t) - \frac{1}{2}|R| + const \tag{4.60}$$

where *R* is the covariance of the observation noise $v_t$, $'$ operator is the matrix transpose, and $|\cdot|$ is the matrix determinant. The model parameters can be estimated by maximising Eq. (4.60). By taking derivatives of Eq. (4.60), one obtains linear system of equations. For example, the ML estimate of the observation matrix *C* can be found from:

$$C \leftarrow \left( \sum_t Y_t X_t' \right) \cdot \left( \sum_t X_t X_t' \right)^{-1} . \tag{4.61}$$

Since the states are in fact hidden, in the *M* step one uses the expected values. The expected value of *f(x)* with respect to the posterior distribution of *X* in shorthand notation is denoted here as $\langle f(X) \rangle$, and can be estimated as:

$$\langle f(X) \rangle = \int_X f(X) P(X|Y, \theta_k) dX . \tag{4.62}$$

Using the expected values, then the *M* step for the observation matrix *C* estimation now becomes:

$$C \leftarrow \left( \sum_t Y_t \langle X_t \rangle' \right) \cdot \left( \sum_t \langle X_t X_t' \rangle \right)^{-1} . \tag{4.63}$$

Similar *M* steps can be derived for all the other parameters by taking derivatives of the expected log probability (Shumway and Stoffer, 1982; Digalakis et al., 1993), which in general requires computation of terms such as $\langle X_t \rangle$, $\langle X_t X_t' \rangle$, $\langle X_t X_{t-1}' \rangle$. These terms can be efficiently computed using *Kalman smoothing* algorithm. The Kalman smoother solves the problem of estimating the state at time *t* of a linear-Gaussian state-space model given the model parameters and a sequence of observations $\{Y_1, \ldots, Y_t, \ldots, Y_T\}$. It basically consists of two parts: a forward recursion which uses the observations from $Y_1$ to $Y_t$, known as *Kalman filtering* (Kalman and Bucy, 1961), and a backward recursion which uses the observations from $Y_T$ to $Y_{t+1}$ (Rauch, 1963), which is known and *Rauch-Tung-Striebel smoother*. We have already previously discussed that in order to compute marginal probability of a variable in Bayesian network, one needs to take into account both the evidence above and below the variable. In fact, the Kalman smoother is simply a special case of the belief propagation algorithm. The Gaussian marginal density of the hidden state vector is completely defined by its mean and covariance matrix. If one denotes the quantities $X_t^\tau$ and $V_t^\tau$ as a mean vector and covariance matrix of $X_t$, respectively, given observations $\{Y_1, \ldots, Y_\tau\}$, the Kalman filter consists of the following forward recursions:

$$X_t^{t-1} = A X_{t-1}^{t-1} \tag{4.64}$$

$$V_t^{t-1} = A V_{t-1}^{t-1} A' + Q \tag{4.65}$$

$$K_t = V_t^{t-1} C' \left( C V_t^{t-1} C' + R \right)^{-1} \tag{4.66}$$

$$X_t^t = X_t^{t-1} + K_t\left(Y_t - CX_t^{t-1}\right) \tag{4.67}$$

$$V_t^t = V_t^{t-1} - K_t C V_t^{t-1} \tag{4.68}$$

where the $X_1^0$ and $V_1^0$ are the prior mean and covariance of the state, which are model parameters. Equations (4.64) and (4.65) in fact describe the forward propagation of the state mean and variance before having seen the observation at time *t*. The mean evolves according to the known dynamics *A* which also affects the variance. In addition, the variance also increases by the state noise *Q*. The observation $Y_t$ has the effect of shifting the mean by an amount proportional to the prediction error $Y_t - CX_t^{t-1}$, where the proportionality term $K_t$ is known as the *Kalman gain matrix*. Observing $Y_t$ also has the effect of reducing the variance of $X_t$. At the end of the forward recursions the values for $X_T^T$ and $V_T^T$ are obtained. One now needs to proceed backwards and evaluate the influence of future observations on the estimates of the states in the past:

$$J_{t-1} = V_{t-1}^{t-1} A'\left(V_t^{t-1}\right)^{-1} \tag{4.69}$$

$$X_{t-1}^T = X_{t-1}^{t-1} + J_{t-1}\left(X_t^T - AX_{t-1}^{t-1}\right) \tag{4.70}$$

$$V_{t-1}^T = V_{t-1}^{t-1} - J_{t-1}\left(V_t^T - V_t^{t-1}\right)J_{t-1}' \tag{4.71}$$

where $J_t$ is a gain matrix similar to the Kalman gain matrix. In addition, one can also recursively compute the covariance across two time steps (after Shumway and Stoffer, 1982):

$$V_{t,t-1}^T = V_t^t J_{t-1}' + J_t\left(V_{t+1,t}^T - AV_t^t\right)J_{t-1}' \tag{4.72}$$

which is initialised as $V_{T,T-1}^T = \left(I - K_T C\right)AV_{T-1}^{T-1}$. Finally, the expectations required for the optimisation algorithm, for example EM, can now be straightforwardly computed as:

$$\left\langle X_t \right\rangle = X_t^T \tag{4.73}$$

$$\left\langle X_t X_t' \right\rangle = X_t^T X_t^{T'} + V_t^T \tag{4.74}$$

$$\left\langle X_t X_{t-1}' \right\rangle = X_t^T X_{t-1}^{T'} + V_{t,t-1}^T \ . \tag{4.75}$$

LEARNING HIDDEN MARKOV MODELS

Recalling the Eq. (4.41), the *log* probability of the hidden variables and observations for an HMM can be written as:

$$log\, P(\{S_t, Y_t\}) = log\, P(S_1) + \sum_{t=1}^{T} log\, P(Y_t|S_t) + \sum_{t=2}^{T} log\, P(S_t|S_{t-1}) \,. \tag{4.76}$$

Let us represent the *K*-valued discrete state $S_t$ using *K*-dimensional unit column vectors, e.g. the state at time *t* taking on the value "2" is represented as $S_t$=[010…0]′. Each of the terms in Eq. (4.76) can be decomposed into summations over the sate variable *S*. For example, the transitional probability is:

$$P(S_t|S_{t-1}) = \prod_{i=1}^{K}\prod_{j=1}^{K} (P_{ij})^{S_{t,i}S_{t-1,j}} \tag{4.77}$$

where $P_{ij}$ is the probability of transition from state *j* to state *i*, arranged in a *K*×*K* matrix P. Then the *log* of transitional probability can be expressed as:

$$log\, P(S_t|S_{t-1}) = \sum_{i=1}^{K}\sum_{j=1}^{K} S_{t,i}S_{t-1,j}\, log\, P_{ij} = S_t'(log\, P)S_{t-1} \tag{4.78}$$

using matrix notation. Similarly for the initialisation, if we assume a vector of initial state probabilities, π, then:

$$log\, P(S_1) = S_1'\, log\, \pi \,. \tag{4.79}$$

Finally, the emission probabilities depend on the type of the observations. In general, we can express the *log* of the emission probabilities as:

$$log\, P(Y_t|S_t) = Y_t'(log\, E)S_t \tag{4.80}$$

where *E* is the emission probability matrix. Since the state variables are hidden, we cannot compute all terms in Eq. (4.76) directly. In this case, the EM algorithm, which for the case of HMMs in known as the Baum-Welch algorithm (Baum et al., 1970), can be used to compute the expectations under the posterior distribution of the hidden states given the observations. These expectations can be expressed as a function of $\langle S_t \rangle$ and $\langle S_t S_{t-1}' \rangle$. The first term $\langle S_t \rangle$ is a vector containing the probability that the HMM was in each of the *K* states at time *t* given its current parameters and the entire sequence ($1 \le t \le T$) of observations. The second term $\langle S_t S_{t-1}' \rangle$ is a matrix containing the joint probability that the HMM was in each of the $K^2$ pairs of states at times $t-1$ and *t* given its current parameters and the entire sequence of observations. In the HMM notation

(Rabiner and Juang, 1986), which is mostly used in the literature, $\langle S_t \rangle$ corresponds to $\gamma_t$ and $\langle S_t S'_{t-1} \rangle$ to $\xi_t$. Given these expectations, the maximisation step is quite straightforward: one takes derivatives of the Eq (4.76) with respect to the model parameters, sets these to zero, and solves subject to sum-to-one constrains that ensure valid transition, emission and initial state probabilities. For example, for the transition matrix one can obtain:

$$P_{ij} = \frac{\sum_{t=2}^{T} \langle S_{t,i} S_{t-1,j} \rangle}{\sum_{t=2}^{T} \langle S_{t-1,j} \rangle} . \tag{4.81}$$

The required expectations are computed using the so-called *forward-backward algorithm*. This algorithm is simply belief propagation applied to the DBN corresponding to HMM (see Smyth et al., 1997). The forward pass recursively computes $\alpha_t$, define as joint probability of $S_t$ and the sequence of observations $Y_1,\ldots, Yt$:

$$\begin{aligned} \alpha_t = P(S_t, Y_1,\ldots,Y_t) &= \left[ \sum_{S_{t-1}} P(S_{t-1}, Y_1,\ldots,Y_{t-1}) P(S_t|S_{t-1}) \right] P(Y_t|S_t) \\ &= \left[ \sum_{S_{t-1}} \alpha_{t-1} P(S_t|S_{t-1}) \right] P(Y_t|S_t). \end{aligned} \tag{4.82}$$

On the other hand, the backward recursive pass computes $\beta_t$, defined as the conditional probability of the observations $Y_{t+1},\ldots, Y_T$ given $S_t$:

$$\begin{aligned} \beta_t = P(Y_{t+1},\ldots,Y_T|S_t) &= \sum_{S_{t+1}} P(Y_{t+2},\ldots,Y_T|S_{t+1}) P(S_{t+1}|S_t) P(Y_{t+1}|S_{t+1}) \\ &= \sum_{S_{t+1}} \beta_{t+1} P(S_{t+1}|S_t) P(Y_{t+1}|S_{t+1}). \end{aligned} \tag{4.83}$$

Having computed $\alpha_t$ and $\beta_t$, the expectations needed for EM are:

$$\langle S_{t,i} \rangle = \gamma_{t,i} = \frac{\alpha_{t,i}\beta_{t,i}}{\sum_j \alpha_{t,j}\beta_{t,j}} \tag{4.84}$$

$$\langle S_{t,i}S_{t-1,j} \rangle = \xi_{tij} = \frac{\alpha_{t-1,j}P_{ij}P(Y_t|S_{t,i})\beta_{t,i}}{\sum_{k,l} \alpha_{t-1,k}P_{kl}P(Y_t|S_{t,l})\beta_{t,l}} \ . \tag{4.85}$$

Note that the Kalman smoothing algorithm and the forward-backward algorithm are conceptually identical. The forward-backward algorithm will be discussed in more detail in the next Chapter 5 where we introduce a hybrid Hidden Markov Mixture of Models (HMMMs) framework for data-driven modelling.

## 4.6 Summary

In general, Bayesian networks are a concise graphical formalism for describing probabilistic models. Dynamic Bayesian networks are designed to handle temporal data. In this chapter we have provided a partial overview of the methods for learning and inference in dynamic Bayesian networks, focussing on the case when the model(s) structure is defined *a priori* using background knowledge of the underlying dynamics of the modelled system. Within the wide range of the interesting DBN models (see Berger, 1999 for a recent overview), we have focused on the general notion of state-space models and HMMs in particular, and have presented how the model parameters can be trained from data.

# Chapter 5
# A Hybrid Framework for Modelling Nonlinear Dynamical Systems

## 5.1 Introduction

Since the advent of cybernetics, nonlinear dynamical systems have been an important modelling tool in various fields ranging from physical to social sciences. Most real dynamical systems have three essential features. First, they show quite irregular dynamical evolution—the observed outputs show very rich and complex dynamical structures in their evolutions, although we try to assume that they are driven by some deterministic dynamics. Second, they have a quite large stochastic component—the observed outputs are noisy nonlinear function of the inputs, and the dynamics itself may be sometimes driven by some unobserved "noise" process. Third, they can be characterised by some finite-dimensional phase space (where some internal variables may not be directly observable) that summarises most of the information about the past behaviour of the underlying processes relevant for identification, classification and prediction of its future evolution.

From a modelling standpoint, as we have already elaborated in Chapter 3, irregularity and chaos are essential to allow a dynamical system with few variables to generate very rich and complex dynamical structures, characterised by the presence of chaotic dynamics, different dynamical regimes (even coexisting attractors) and an irregular dynamical evolution between them. These irregular (stochastic-like) transitions between different dynamical regimes of the system can be modelled using dynamic Bayesian networks, such as state-space and hidden Markov models, as discussed in Chapter 4. In addition, existence of such transitions between different dynamic regimes is a source of non-stationarities, which are severe problem in modelling dynamical systems.

A basic hybrid framework for modelling such complex nonlinear dynamical systems is the *mixture of experts* (ME) framework, introduced by Jacobs at al. (1991) in the neural network community. The mixture of experts framework, which comes also by many other names such as modular networks or multiple models or ensemble of models, aims at separating the seemingly complex global dynamics into a couple of lower-dimensional sub-dynamics which can be modelled by separate models (experts) more easily. In terms of the language of deterministic chaos, it means that the separate models (experts) specialise on modelling different parts of the reconstructed phase space based on the different geometrical complexities of the attractor. For example, the Lorenz dynamical system (see Section 3.1) exhibits switching between two different oscillatory modes (dynamic regimes) which are globally nonlinear. Instead of modelling the dynamical structure by a "global" or "monoclined" models, one could think of employing two simple models (even linear as we will demonstrate latter) that specialise on the two

different oscillatory modes, and then the nonlinearity can be incorporated into the gating procedure. A central problem of using a mixture of models framework is therefore the calculation of the activation of each model (expert), called the *gating problem*.

Figure 5.1 shows the architecture of a hybrid ME framework, consisting of three experts and one gating model both having access to the input vector. Note that in our case the input space is the reconstructed phase space of the dynamical system based on the time series of the observables.

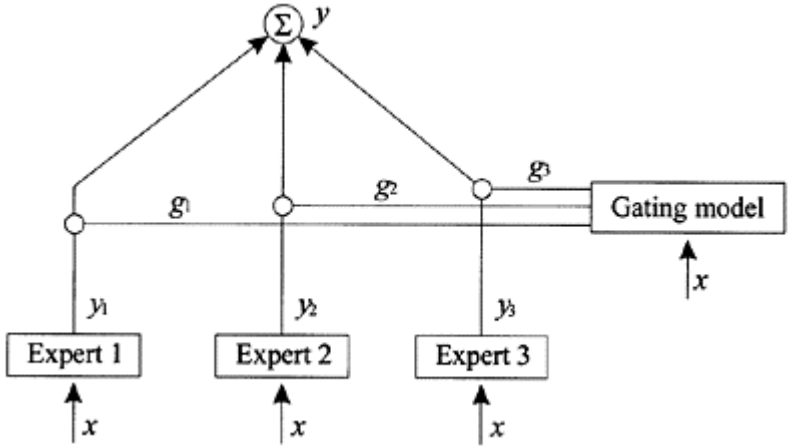



Figure 5.1. Architecture of a hybrid mixture of models framework

The gating model has one output $g_j$ per expert (model). The output of the ME is the weighted (by the gating model outputs) mean of the expert outputs:

$$y(x)=\sum_{j=1}^{m} g_j(x)\cdot y_j(x)\ . \qquad (5.1)$$

The gating model of the ME learns in fact to partition the input space (phase space) in a soft way, and attributes experts to these different regions. In particular, the gating model outputs $g_j(x)$ can be regarded as the conditional probability that a particular expert has generated the output given the data set and the model structure. A probabilistic interpretation of the ME framework is recently described in Boshop (1995, see section 6.4).

In the last decade, several solutions have been proposed to the gating problem of the hybrid ME framework. In its original formulation (Jacobs et al., 1991) the expert activities are provided by a feed-forward gating neural network based on the input vector. Further extension by the use of a recurrent gating network, in order to take into account the previous performances of the experts, is elaborated by Cacciatore and Nowlan (1994).

Kehagias and Petridis (1997) proposed an algorithm for on-line time series segmentation using predictive modular neural networks. An alternative, non-recurrent

approach to distinguish between different dynamic regimes is the *annealed competition of experts* (ACE) method proposed by Pawelzik et al. (1996). It has its roots in statistical mechanics and is a purely performance-driven concept, which considers a moving average prediction error for estimating the experts' activities.

In contrast to these approaches, we use the concept of Hidden Markov Mixture of Models (Experts)—HMMMs and associate each prediction model (expert) with a hidden state of the system corresponding to a particular dynamic regime. This concept is similar to the concept of input-output hidden Markov models introduced by Bengio and Frasconi (1995). However, the proposed approach can be seen as more general and conceptually more flexible by integrating: (i) the input information; (ii) the information on the position in the phase space and previous evolutions; (iii) the performance of the models that are particularly learned and suited to model different dynamical regimes; and (iv) information on the hidden dynamical regimes for modelling the gating probabilities by HMM. The proposed hybrid HMMMs framework for modelling nonlinear dynamical systems is further elaborated mathematically, demonstrated and discussed in this chapter.

## 5.2 Hidden Markov mixture of models (experts)

### *5.2.1 Description and underlying assumptions*

The proposed hybrid HMMMs framework is a combination of both, the modelling of the reconstructed phase space of nonlinear dynamical systems, and the dynamic Bayesian networks expressed with hidden Markov models throughout the gating procedure. It is best described by the following underlying assumptions:

- There are several discrete hidden states that we call dynamic regimes. Each of these dynamic regimes is modelled by a corresponding mapping function in the reconstructed phase space of the dynamical system. These models are called experts, and they can range from local models to neural networks, and even to conceptual models. Note that any input-output mapping can be modelled, not necessarily the reconstructed phase space.
- At each time step a single expert, modelling a particular dynamic regime, is responsible for generating the corresponding observation. We do not know which of the experts actually generated the observation; thus the posterior emission probabilities of the experts for each time step need to be estimated from the data.
- For modelling the sequence of the hidden dynamic regimes, we assume that the dynamics of these regimes is quite irregular and can be described by a first order Markov process. Thus, the next dynamic regime depends on the current dynamic regime and implicitly on the data. This is expressed as a matrix of transitional probabilities between the hidden dynamic regimes (recall HMMs from the previous chapter). We do not know these transitional probabilities. The initial distribution can be assign by an expert depending on the modelling problem, but the probabilities will need to be adjusted (estimated) from the data.

In this section we further present how we can learn the model parameters (if necessary) for each expert, the parameters of the transition matrix, and the emission probability

vector for each expert across the hidden dynamical regimes at each time step. The process of learning builds on the forward-backward estimation procedure and the EM algorithm previously described in Chapter 4. The description of this proposed framework includes the following characteristics and implications:

(i) *Discovering hidden dynamical regimes.* Intelligent data analysis and data mining in general often use the term "discover hidden knowledge", but without clearly defining the concept "hidden knowledge". This framework clearly defines hidden dynamical regimes as the components of the mixture density. The presented Bayesian probabilistic approach allows for a principled interpretation in terms of probabilities, enabling the discovery of interesting relations. For example, in the case of predicting the runoff, the hidden regimes can be referred to as "catchment preparation" or "seasonal effects" depending on the considered time scale; in case of surge water level prediction, the hidden regimes can be referred to as a "storm phases" for example. Methodologically, it is important to clarify that this approach does not insert knowledge that "catchment preparation" is important for the peak runoffs, but it does make assumptions that in turn yield this knowledge.

(ii) *Combining supervised with unsupervised learning.* Approaches to learning from data are traditionally divided into supervised and unsupervised learning. The proposed hybrid framework combines the strengths of both approaches to learning: the advantage of supervised learning constraining the model structure, parameters and performance evaluation, while providing the flexibility of unsupervised learning that allows for discovering and interpreting the sequence of the hidden dynamical regimes.

(iii) *Becoming experts through competition.* This framework uses competitive learning, meaning that for each training pattern, all experts compete. If a particular expert's prediction is better that the others, it receives a larger share of the data point to update its parameters than the others. Thus, it learns to improve its predictions in the areas where it is already good, and learns to ignore areas where some other experts are better.

(iv) *Combining forecasts.* The idea of combining forecasts, going back to Bates and Granger (1969), has become increasingly important in applied forecasting and especially when different models produce different qualitative and quantitative forecasts. In most approaches to forecast combination, the individual models are given equal weights to all their training data points. This framework allows for soft combination of the forecasts of the experts where the relative weights of each expert vary at each time step. These weights are the estimates of the posterior probabilities and they reflect the training set performance of the experts in similar situations.

(v) *Coping with outliers.* Many practical problems in data-driven modelling use some heuristic to remove outliers. Given the strong effect that outliers in general have on the learned model, this heuristic in dealing with outliers can significantly determine the model capabilities. As an alternative to removing outliers, robust statistics uses an influence function that downweights patterns where there is a big discrepancy between the predictions and the observations. However this approach is not applicable in the area of risk modelling which focuses on rare events and on tails of distributions. In contrast, this framework can cope with outliers in a way that one (or more) expert can be designed with a proper capacity and structure (large variance) in comparison with the other experts. The role of this expert (or experts) will be to act as a "garbage-

collector", effectively removing the outliers and "specialising" on them much better than the other experts. In turn, the other experts will not be affected with the big discrepancies in the predicted and observed patterns.

### *5.2.2 Basic architecture of the HMMMs framework*

The basic architecture of the proposed HMMMs framework with the used notation is presented in Figure 5.2 and described bellow.

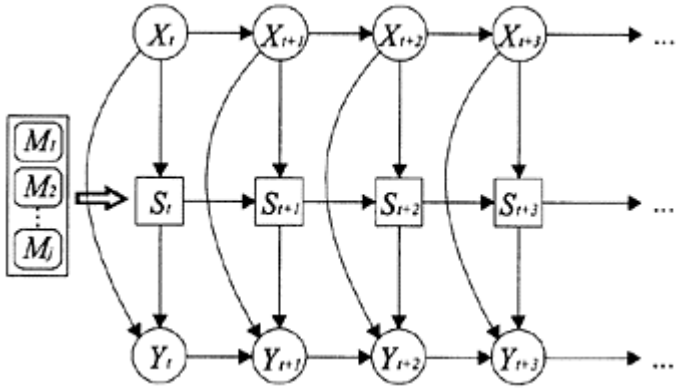



Figure 5.2. The architecture of the HMMMs framework.

NOTATION

1. *Observations:* $Y^T=\{y^t|t=1,\ldots, T\}$ refers to the observed time series data. $T$ is the number of the observations and $t$ is the time index. Similarly, $X^T=\{x^t|=1,\ldots, T\}$ represents the input to the emission model. The input data $x^t$ itself can be a vector (multivariate time series) or a scalar. In case of phase-space modelling, $x$ is given by the previous lagged values, $x^t=\{y^t, y^{t-\tau}, y^{t-2\tau},\ldots, y^{t-m\tau}\}$, where $m$ is the embedded dimension and $\tau$ is proper time delay of the embedded input properly estimated as explained in Chapter 3.
2. *Dynamic regimes (states):* $S=\{1, 2,\ldots, j,\ldots, M\}$ denotes the dynamic regime (state) or an expert. $M$ is the number of dynamic regimes in the system and $j$ refers to a specific dynamic regime which is modelled by a particular model—expert. Note that the modes for a particular dynamic regime could range from global to local and from linear to nonlinear models (e.g. neural networks).
3. *Transition probabilities (gating procedure):* $a_{ij}$ is the transition probability of switching from dynamic regime $i$ to $j$:

$$\mathbf{A}=\{a_{ij},\ i, j\leq M,\ a_{ij}=P(s^{t+1}=j|s^t=i)\} \tag{5.2}$$

$\sum_j a_{ij} = 1$

where $a_{ij} \geq 0$, and $s^t$ describes the dynamic regime (state) in time $t$.

4. *Emission probabilities:* $b_{ij}$ is the probability of observing $y^t$ given the state and the model. In the HMME this probability depends on the inputs $x^t$ into the experts at time $t$ through the conditional mean:

$$\mathbf{B} = \left\{ b_j^t, \;\; j \leq M, \;\; t \leq T, \;\; b_j^t = P\left( y^t \middle| s^t = j, x^t \right) \right\} \tag{5.3}$$

5. *Initial probabilities of each state:* $\prod = \{\pi_i, i=1,\ldots, M\}$, where the probabilities have sum to unity, $\sum_i^M \pi_i = 1$.

For convenience, $\theta = \{\mathbf{A}, \mathbf{B}, \prod\}$ denotes the entire set of parameters of the model. The emission probability can thus be written as $b_j^t = P\left( y^t \middle| s^t, x^t, \theta \right)$.

## DEFINING THE LIKELIHOOD FUNCTION

In order to estimate the most likely path (sequence) of the dynamic evolution of the system, given the input data and the observations (output) generated by the system being throughout different dynamic regimes, one needs to define the likelihood function. On the basis on the conditional independence assumption for the gating mechanism, the probability of the current state depends only on the previous state (1st order Hidden Markov Model):

$$P(s^t | s^{t-1}, s^{t-2}, \ldots, s^1, X^{t-1}, Y^{T-1}) = P(s^t | s^{t-1}) \tag{5.4}$$

Denoting the specific sequence or path of the dynamic evolution of the system from $t=1$ to $T$ as $q^T$, this first order Markov assumption for the gating mechanism allowing to write the probability of the path $q^T = (s_1, s_2, \ldots, s_T)$ as:

$$P\left(q^T\right) = P\left(s^1, s^2, \ldots, s^T\right) = P(s^1) \prod_{t=2}^{T} P\left( s^t \middle| s^{t-1} \right) \tag{5.5}$$

Given the current input pattern $x^t$, output $y^t$ and the previous state $s^{t-1}$, earlier values of $s$ and $y$ are irrelevant, thus:

$$P(y^t, s^t | q^{t-1}, X^{t-1}, Y^{t-1}) = P(y^t, s^t | s^{t-1}, x^t) \tag{5.6}$$

Using Eq. (5.4) this expression can be transformed in the following way:

$$P(y^t, s^t | s^{t-1}, x^t) = P(y^t | s^t, x^t) P(s^t | s^{t-1}) \tag{5.7}$$

The central problem of the hidden Markov models is to find (learn) the parameters of the model. Using the Eq. (5.6) and Eq. (5.7), the likelihood of generating the observables given the input and model parameters $P(Y^T|\theta)$ can be expressed as:

$$P\left(Y^T\middle|\theta\right)=\sum_{q^T}P\left(Y^T,q^T\middle|\theta\right)=\sum_{q^T}P\left(y^T,s^T\middle|q^{T-1},Y^{T-1},\theta\right)P\left(Y^{T-1},q^{T-1}\middle|\theta\right) \tag{5.8}$$

Using the Eq. (5.6) the likelihood becomes:

$$P\left(Y^T\middle|\theta\right)=\sum_{q^T}P\left(y^T,s^T\middle|s^{T-1},x^T,\theta\right)P\left(Y^{T-1},q^{T-1}\middle|\theta\right) \tag{5.9}$$

Furthermore using the Eq. (5.7) the likelihood can be written as:

$$P\left(Y^T\middle|\theta\right)=\sum_{q^T}P\left(y^T\middle|s^T,x^T,\theta\right)P\left(s^T\middle|s^{T-1}\right)P\left(Y^{T-1},q^{T-1}\middle|\theta\right) \tag{5.10}$$

Finally, the likelihood can be expressed in the following form:

$$P\left(Y^T\middle|\theta\right)=\sum_{q^T}P\left(y^1\middle|s^1,x^1,\theta\right)P\left(s^1\right)\prod_{t=2}^{T}P\left(y^t\middle|s^t,x^T,\theta\right)P\left(s^t\middle|s^{t-1}\right) \tag{5.11}$$

$b^1$ $s^1$ $b_j^t$ $a_{ij}$

In order to compute the likelihood, two probabilities need to be estimated. First, the emission probability given the current dynamic regime (state), $P(y^t|s^t, x^T, \theta)$, which varies at each time step. Second, the transition probability $P(s^t\, s^{t-1})$, which is a parameter of the model. The product of the last terms $b_j^t \cdot a_{ij}$ in the Eq. (5.11) is at the heart of the hidden Markov gating framework applied in this approach. Without the Markov assumption, the second term $a_{ij}$ is absent and the observation (output of the dynamical system) at time $t$ would be attributed to regime (state) $j$ with probability $b_j^t \Big/ \sum_{i=1}^{M} b_i^t$. This type of model-based time series clustering (without the term $a_{ij}$) is the unconditional case (no input pattern $x$). The presence of the second term $a_{ij}$ introduces the trade-off with the first term towards the entire likelihood. In most of the natural dynamical systems, the diagonal elements of the transitional probabilities $a_{ii,}$ describing the self-transitions (i.e. probability of staying in a dynamic regime), typically have high values (above 0.90), indicating persistency. This means that the system will switch to other dynamic regime (state) if the next data patterns (forcing into the system) and the previous sequence (path) of the dynamic evolution of the system can be explained much better by a dynamic regime different from the current one.

## MODELLING THE CONDITIONAL EMISSION PROBABILITIES

The main assumptions used in estimation of the emission probabilities $b_j^t$ by each of the employed models for the different dynamic regimes are:

*Independence:* Given the parameters of the emission model, the likelihood of observing $y^t$ given the current dynamic regime (state) and the given input pattern, is $b_j^t = P\left(y^t \middle| s^t = j, x^t, \theta\right)$. The emission probabilities are independent for each time step *t*.

Each of the specified emission model is known as expert, and each individual expert is responsible for modelling a particular dynamic regime.

*Density function:* The framework presented here allows for the assumption of different density distribution for the error function or the "noise". In the specific example of a Gaussian density distribution, the emission probability of expert (or model) *j* becomes:

$$b_j^t = P\left(y^t \middle| s^t = j, x^t, \theta\right) = \sqrt{\frac{1}{2\pi\sigma_j^2}} \exp\left(-\frac{\left(y^t - \hat{y}_j^t x^t\right)^2}{2\sigma_j^2}\right) \qquad (5.12)$$

where $\hat{y}_j^t x^t$ is the conditional mean and $\sigma_j^2$ is the variance of the predicted Gaussian density.

*The functional form of the experts (models):* The functional dependence of the conditional mean $\hat{y}_j^t x^t$ on its input can be potentially any linear on nonlinear mapping function, such as a radial basis function, a multi-layered perception, wavelet networks etc. In this particular case we use local models based on the concept of local modelling in deterministic chaos. Each of the models (experts) is characterised by different model parameters (such as order of model, number of neighbours, embedding dimension and time delay). Global models can be used as well. The emission probability **B** is determined by the set of parameters $\theta_j$ of expert *j*, according to the architecture of the emission model. Furthermore, the different experts can have different input data sets. Typically, the number of inputs to each expert (model) can be a subset of the full set of inputs identified as important for a description of the dynamics of the system. When different dynamic regimes modelled by the different experts "live" on sub-areas of the reconstructed phase-space of the system, this approach can help to reduce the curse of dimensionality.

## COMPUTING THE LIKELIHOOD

Computing the likelihood $P(Y^T|\theta)$ directly from Eq. (5.11) is intractable. As already discussed in the previous Chapter 4, Baum (1970) proposed an elegant method called the *forward-backward* algorithm. Dempster (1997) subsequently introduced the so-called Expectation-Maximisation or EM algorithm to maximise this probability due to the presence of the hidden variables. The proposed HMMM framework builds on these two algorithms.

*The forward-backward algorithm applied*

Let $\alpha_i^t$ the *joint probability* of having observed *y* from time 1 to *t* and of being in dynamic regime *i* at time *t*:

$$\alpha_i^t = P\left(y^1, y^2, ..., y^t, s^t = i \middle| \theta\right) \tag{5.13}$$

where $1 \le t \le T$ and θ denotes the model parameters. The probability of the *entire sequence of observations* is given by the sum over the dynamic regimes (states) at the end of the sequence (at time *T*):

$$P\left(Y \middle| \theta\right) = \sum_{i=1}^{M} \alpha_i^t \tag{5.14}$$

As already discussed, the nice point of this algorithm is the decrease of computational complexity. Rather than being exponential in time (given the consideration of all possible paths), this computation is only linear in time, since $\alpha_i^t$ can be computer recursively:

$$\alpha_i^{t+1} = \left( \sum_{i=1}^{M} \alpha_i^t a_{ij} \right) \cdot b_j^{t+1} \tag{5.15}$$

At the beginning of the sequence, the variable $\alpha_i^t$ are initialised with probability $\alpha_i^1 = \pi_i b_i^1$. This recursion is called the forward procedure. Given initial estimates of $\pi_i$ and $b_i^1$, Eq. (5.15) allows the computation of the probability $P(Y^T|\theta)$, and for $t=T$, the entire likelihood. Similarly, the backward variable $\beta_i^t$ is defined as the conditional probability of observing *y* from *t+1* to *T* given the dynamic regime *i* at time *t* and the model parameters:

$$\beta_i^t = P\left(y^{t+1}, y^{t+2}, ..., y^T \middle| s^t = i, \theta\right) \tag{5.16}$$

With *t=T−1, T−2,…, 2, 1* one can obtain all $\beta_i^t$ for all *t* in the backward procedure. Combining the variables $\alpha_i^t$ and $\beta_i^t$, it is possible to estimate the *important posterior probability* of being in a dynamic regime *i* at time *t* given the entire set of observations and parameters:

$$\gamma_i^t = P\left(s^t = i \middle| Y, \theta\right) = \frac{P\left(Y, s^t = i \middle| \theta\right)}{P\left(Y \middle| \theta\right)} = \frac{\alpha_i^t \cdot \beta_i^t}{\sum_{k=1}^{M} \alpha_k^t \cdot \beta_k^t} \tag{5.17}$$

The variable $\gamma_i^t$ is a key quantity, which describes the activation of the experts within the entire observed sequence that are responsible for modelling of particular dynamic regimes and will be referred to as an *expert activation function*.

Finally, the *joint probability* of the transition between the dynamic regimes, $P(s^t=i, s^{t+1}=j|Y, \theta)$, can be also computed using the forward-backward procedure:

$$\xi_{ij}^{t,t+1} = \frac{P\left(s^t = i, s^{t+1} = j, Y\middle|\theta\right)}{P\left(Y\middle|\theta\right)} = \frac{\alpha_i^t \cdot a_{ij} \cdot b_j^{t+1}\beta_j^{t+1}}{\sum_{i=1}^{M}\sum_{j=1}^{M}\alpha_i^t \cdot a_{ij} \cdot b_j^{t+1}\beta_j^{t+1}} \tag{5.18}$$

This variable $\xi_{ij}^{t,t+1}$ serves as an additional quantity in the computation of the transitional probabilities.

*The EM algorithm applied*

As mentioned previously, the likelihood as given by Eq. (5.11) cannot be maximised directly since the hidden states are not known. The solution to this problem goes back to Dempster et. al (1977). For set of parameters $\theta$ and $\theta_{old}$, an auxiliary *Q*-function is defined as:

$$Q(\theta, \theta_{old}) = \sum_{\forall q^T} P\left(y^T, q^T\middle|\theta_{old}\right)\log P\left(y^T, q^T\middle|\theta\right) \tag{5.19}$$

It can be shown that $Q(\theta, \theta_{old}) > Q(\theta_{old}, \theta_{old}) \Rightarrow P\left(Y^T\middle|\theta\right) > P\left(Y^T\middle|\theta_{old}\right)$ (Baum et. al, 1970; Dempster et. al, 1977). This re-estimation algorithm is known in the literature as *Baum-Welch* algorithm, which for HMM is known as the Expectation Maximisation algorithm. Its key idea is to alternate between two steps, the E-step and the M-step.

- The E-step (Expectation step) assumes that the parameters of the model are known, and computes for each time step t the variables $\alpha_i^t$ and $\beta_i^t$, and in turn the posterior probabilities $\gamma_i^t$ and $\xi_{ij}^{t,t+1}$.
- The M-step (maximisation step) takes the variables computed in the E-step and updates the parameters of the model such that Eq. (5.19) is maximised under the constrains $\sum_{i=1}^{M}\pi_i = 1$ and $\sum_{j=1}^{M}a_{ij} = 1$.

The new *transitional probabilities* are estimated as:

$$a_{ij} = \frac{\text{expected number of transitioins from state } i \text{ to } j}{\text{expected number of transitions from state } i \text{ (to anywhere)}} = \frac{\sum_t \xi_{ij}^{t,t+1}}{\sum_t \gamma_i^t} \tag{5.20}$$

The new *initial probabilities* of state *i* are $\pi_i = \gamma_i^1$. The formulation for the re-estimation of the emission parameters depend both on the specified error function ("noise model") and the specific functional form for the parameters of the model for each expert (e.g. global linear AR model, nonlinear neural network, local models etc.). For each expert, Eq. (5.19) is maximised when the following *G*-function is maximised (Fraser and Dimitradis, 1994):

$$G = \sum_{t=1}^{T} \sum_{j=1}^{M} \gamma_j^t \log P\left(y^t \middle| x^t, s^t = j, \theta_j\right) \tag{5.21}$$

where $\theta_j$ represents the parameters of the emission model of state *j*. Eq. (5.21) can be interpreted as the negative of a cost function for the emission model. The estimation of the parameters $\theta_j$ depends on the specific form of the emission model. In order to be able to mathematically demonstrate the updating of the parameters, the error density function is assumed to be Gaussian, thus each emission model for the expert has two parameters: the conditional mean and the variance. Assuming that the variance $\sigma_j^2$ depends only on the performance on the expert, the likelihood is maximised when the partial derivative:

$$\frac{\partial G}{\partial \sigma_j^2} = \sum_{t=1}^{T} \gamma_j^t \frac{1}{P\left(y^t \middle| x^t, s^t = j, \theta_j\right)} \frac{\partial P\left(y^t \middle| x^t, s^t = j, \theta_j\right)}{\partial \sigma_j^2} \tag{5.22}$$

takes value of zero, yielding:

$$\sigma_j^2 = \frac{\sum_{t=1}^{T} \gamma_j^t \left(y^t - \hat{y}_j^t\right)^2}{\sum_{t=1}^{T} \gamma_j^t} \tag{5.23}$$

This is in fact the $\gamma_j^t$-weighted error between the observation $y^t$ and the predictions $\hat{y}_j^t$. In other words, it describes the local "noise" level for particular expert *j*.

The mean of expert *j*, $[\hat{y}]_j^t(x_j^t)$ is a function of the inputs into the expert (model for particular dynamic regime), $x_j^t$. This, in general, nonlinear dependence is parameterised with $\theta_j$. In order to maximise Eq. (5.21) its partial derivative with respect to the parameters $\theta_j$ has to vanish:

$$\frac{\partial G}{\partial \theta_j} = \sum_{t=1}^{T} \gamma_j^t \frac{1}{P\left(y^t \middle| x^t, s^t = j, \theta_j\right)} \frac{\partial P\left(y^t \middle| x^t, s^t = j, \theta_j\right)}{\partial \theta_j} = \tag{5.24}$$

$$= \sum_{t=1}^{T} \gamma_j^t \frac{y^t - \hat{y}_j^t}{\sigma_j^2} \frac{\partial \hat{y}_j^t}{\partial \theta_j}$$

In the general nonlinear case, each pattern still has the importance $\gamma_j^t$, but the parameters can be estimated iteratively, as an additional inner loop within each M-step. For example, in case of neural networks for each experts and interpreting Eq. (5.23) as a cost function, each expert minimises the weighted squared error:

$$\sum_{t=1}^{T} \gamma_j^t \left(y^t - \hat{y}_j^t(x_j^t, \theta_j)\right)^2 \tag{5.25}$$

The parameters- weights $\theta_j$ can be estimated using standard backpropagation algorithm (see Section 2.??). In case when each expert consist of local models in phase space, the parameters $\theta_j$ are the time delay $\tau$, embedding dimension *m*, the number of nearest neighbours *k* and the order of the model *n*. The expert activation function $\gamma_i^t$ in this case can be viewed as an *effective learning rate*.

## GENERATING THE PREDICTIONS

Many forecast methods (in particular almost all nonlinear forecasting methods) focus on predicting the next value or a point of the time series (discharge in this case). However, the presented framework allows for density predication as well, which may be used for assessing the certainty or uncertainty of the predictions. Having estimated the expert activation function $\gamma_i^t$, the density for $y_j^{t+1}$ can be expressed as a linear superposition of the densities of the individual experts:

$$P\left(y^{t+1} \middle| X^t, Y^t, \theta\right) = \sum_{j=1}^{M} \gamma_j^{t+1} \cdot P\left(y^{t+1} \middle| X^t, s^{t+1} = j, \theta_j\right) \tag{5.26}$$

The point prediction or the overall mean of the predicted density at time *t+1* can be estimated as $\gamma_i^t$-weighted superposition of the individual means:

$$\hat{y}^{t+1} = \sum_{j=1}^{M} \gamma_j^{t+1} \cdot \hat{y}_j^{t+1} \tag{5.27}$$

### *5.2.3 Performance measures*

There are many different criteria that one can use for evaluation of the model performance, which usually depend on the modelling purpose and objective (see for example Hall, 2000). In general, these criteria can be categorised as: graphical and numerical performance indicators. For evaluating the performance of the model, we use the following criteria selected from the graphical indictors groups:

- A scale plot of the simulated or predicted and observed time series for both training and testing periods.
- A frequency distribution and scatter plot of the simulated versus observed flows for the testing period.
- Other various plots of the model performances (such as learning rate, experts activation functions, standard deviation plots etc.).

Visual inspection of plots that compare the prediction of actual measurements can provide significant information about how close the predictions are to the observation for different flow regimes.

From the several numerical indicators, we use the following performance measures:

1) Mean square error (MSE) and Normalised mean square error (NMSE)

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Yobs_i - Ypred_i)^2 \tag{5.28}$$

where *Yobs* is the observed value, *Ypred* is the predicted or computed value, and *N* is the length of time series. MSE measures the average sum of the square of the errors throughout the length of time series. However the MSE is not a good indicator for the error measurements for some classes of problems with quite variable dynamics. The normalised mean square error (NMSE), which is dimensionless quantity, provides better indicator for the error measurement since it is normalised by the variance of the observed data:

*NMSE=MSE/Variance* (5.29)

2) Root mean square error (RMSE) and Normalised root mean square error (NRMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Yobs_i - Ypred_i)^2} \tag{5.30}$$

Indeed, RMSE is the root of MSE and has the same unit as the measurement unit of the variable (water level, discharge etc.). Similarly, NRMSE is obtained by the normalisation of RMSE by the standard deviation of the observed data, i.e.

*NRMSE=RMSE/Standard deviation* (5.31)

3) Coefficient of correlation *(r)* and Coefficient of determination *(D)*

$$r = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(Yobs_i - \overline{Yobs}\right)\left(Ypred_i - \overline{Ypred}\right)}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Yobs_i - \overline{Yobs}\right)^2}\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Ypred_i - \overline{Ypred}\right)^2}} \tag{5.32}$$

where $\overline{Yobs}$ is the mean value of the observed time series and $\overline{Ypred}$ is the mean value of the predicted time series respectively. The coefficient of correlation *r* measures how well the predicted values linearly correlate with the observed one. The ideal value for *r* is 1 when there is a perfect prediction. The coefficient of determination *D* is the square of the coefficient of correlation *r* and measures the variability in the observed and predicted values. Also the ideal value for *D* is 1.

As stressed before, the selection of the criteria highly depends upon the modelling purpose. For example, if one tries to build a model for flood forecasting, then the magnitude of the peak flow and time of occurrence of the peak flow would be important criteria. In such cases, the model performance on these terms is more important than the overall global error measurement, as the global error statistics provide relevant information on overall performance but do not provide specific information about model performance at extreme events. One could then use numerical indicators about the model performance defined over particular thresholds related to different dynamical regimes.

4) Mean absolute error (MAE)

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\left(Yobs_i - Ypred_i\right)\right| \tag{5.33}$$

*MAE* can be, for example, calculated for low-flow and high-flow regimes.

5) Maximum absolute error (MaxAbsErr)

$$MaxAbsErr = \max\left|\left(Yobs_i - Ypred_i\right)\right| \tag{5.34}$$

These measures in different dynamical regimes, in combination with the global statistics, provide a better insight into the performance of the model.

### 5.3 Testing the HMMMs framework

The algorithm for the HMMMs framework was implemented and tested in the MatLab computational environment. Several tests were carried out using synthetic time series generated by known dynamical system, both linear and nonlinear. For complex dynamical systems, it is important to analyse and understand the behaviour of this hybrid framework and build up some modelling heuristics in cases when the model assumptions deviate from those of the generating process. Since the proposed framework contains an unsupervised part in the learning from data, this section investigates whether the dynamic

regimes found by the hybrid model actually correspond to the true hidden states modelled by the individual experts. Two types of models (experts) were used in the experiments, namely local models and global linear models. One-step-ahead forecasts were further compared with two architectures of neural networks: multilayered perceptron and modular neural network. The following cases, reflecting wide range of different dynamical systems were tested:

1. Time series generated by dynamical system exhibiting three different regimes corresponding to the Markov chain of hidden states;
2. Time series generated by a dynamical system with two regimes whose sub-dynamics is driven by autoregressive process of order 2 (polluted with noise), and reflecting periodic nonstationarity;
3. Time series generated by two completely random models reflecting pure stochastic process;
4. Time series generated by nonlinear deterministic chaos: Lorenz model and McKay-Glass model.

### *5.3.1 Linear models with Markov chain of hidden states*

The data generation for this experiment consists of two distinct and different processes: the Markov chain of hidden dynamic regimes, and the dynamics of the individual experts.

• *Dynamic of the hidden regimes:* Three hidden regimes were used to generate the underlying dynamics of the system. The transitional probabilities used to describe the Markov model between the three experts are given by the matrix:

$$A = \begin{pmatrix} 0.81 & 0.10 & 0.09 \\ 0.21 & 0.71 & 0.08 \\ 0.03 & 0.06 & 0.91 \end{pmatrix}$$

This allows to generate a realisation for the times series of the hidden dynamical regimes.

• *Dynamics of the individual experts:* Each individual expert were rather simple multivariate global linear models polluted by additive white noise ε, written in a form as:

$$Y^t = \begin{cases} 0.2X_1^t + 0.6X_2^t + \varepsilon_1 & \text{if in regime 1} \\ 0.4X_1^t + 0.3X_2^t + \varepsilon_2 & \text{if in regime 2} \\ 0.55X_1^t + 0.15X_2^t + \varepsilon_3 & \text{if in regime 3} \end{cases}.$$

We have used two inputs $X_1$ and $X_2$ with the following matrix of coefficients $b$=[0.2 0.6; 0.4 0.3; 0.55 0.15] depending in which dynamic regime is the system. The inputs $X_1$ and $X_2$ were artificially correlated with the output $Y$ using *sine* and

*cosine* functions, so that the correlation coefficients were 0.68 and 0.71 respectively. The output time series was polluted with 2% zero-mean white noise.

We first generated a sequence of length 2000 of the (eventually hidden) regimes. This sequence determined which of the three processes was used for each time step to generate an observation $Y^t$. Although the underlying dynamics of the individual experts is simple, the total dynamic of the system becomes intrinsically complex. The recognition models for the individual experts were global multivariate linear models and the learning task was to learn the model parameters and the parameters of the gating procedure. From the generated data we used the first 1600 samples for training and last 400 samples for cross-validation and testing respectively (200 samples each). Figure 5.3 shows both the expert activation functions used to generate the time series and the uncovered expert activation functions from the HMMMs framework respectively.

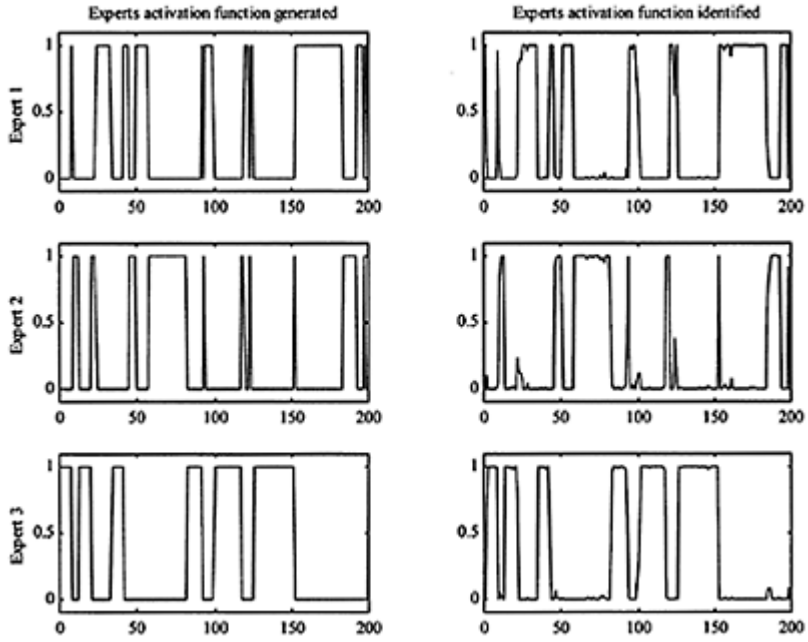



Figure 5.3. Generated (left) and recognised (right) sequence of the activation functions for the three experts. The figure shows only the first 200 samples from the time series.

Form the From the Figure 5.3 is it evident that the framework is able to accurately identify the expert activation functions for each expert. Figure 5.4 shows the complete time series of the activation functions of the three experts and the time series of the observable.

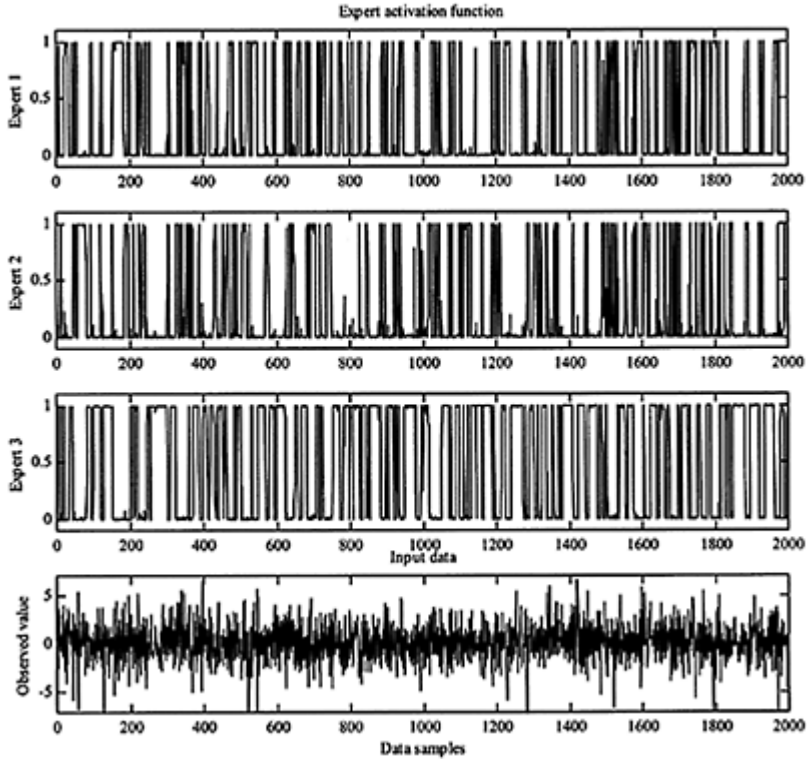



Figure 5.4. Activation of the three experts showing “specialisation “in different dynamic regimes.

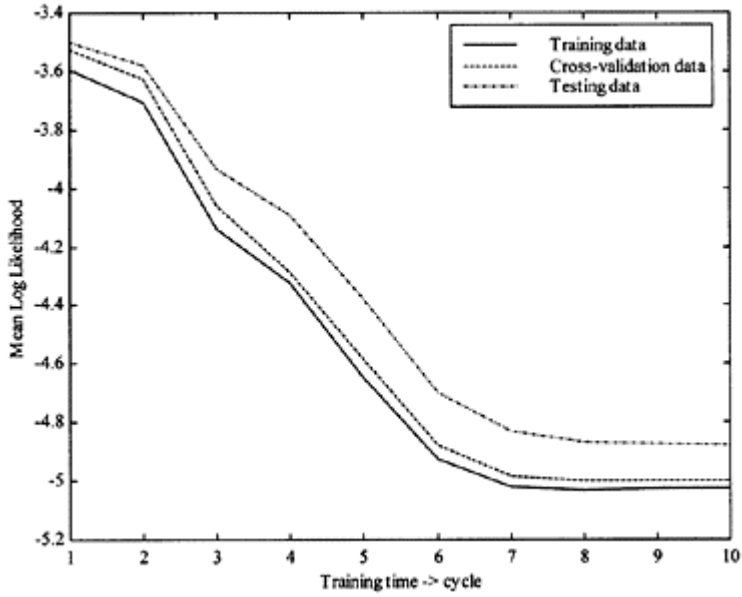



Figure 5.5. Performance of the model by datasets.

Figure 5.5 shows the performance of the model for the training, cross-validation and testing datasets over ten EM cycles.

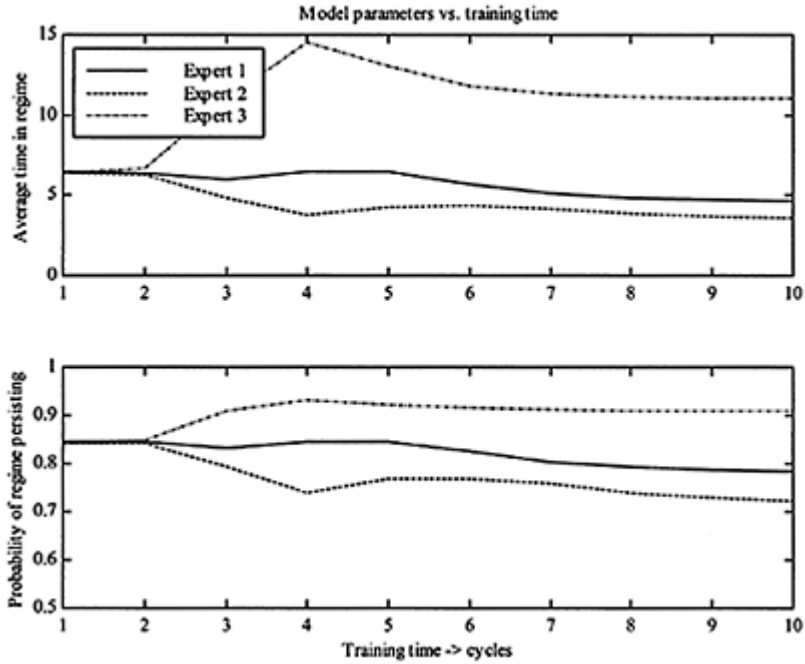



Figure 5.6. Transitional probabilities of regimes persisting for the experts.

The transitional probabilities of staying in the same dynamic regimes with the average time in regimes for the experts during the training cycles are presented in Figure 5.6. The results indicate that starting from the initial transitional probabilities of regime persisting of 0.85 for each expert, during the training stage the frameworks is capable of learning the correct transitional probabilities used to generate the synthetic time series initially.

The predictive performances of the model for the three data sets were evaluated using the performance indicators summarised in Table 5.1, and using scatter plots and histograms as shown in Figure 5.7 and Figure 5.8. These results were further compared to the multilayered perceptron using architecture 2×4×1 with *tanhyp* transfer functions in the hidden layer and linear transfer function at the output, and a modular neural network using three separate MPLs with similar architecture. Figure 5.9 further shows the observed and the predicted values of the testing set using HMMMs framework and the neural network models.

Table 5.1. Modelling error for the testing data set using HMMMs, MPL and modular NN.

| Model | NMSE | RMSE | NRMSE | *r* | *D* |
|---|---|---|---|---|---|
| HMMMs | 0.011 | 0.103 | 0.059 | 0.998 | 0.996 |
| Multi layered perceptron | 0.150 | 0.674 | 0.386 | 0.922 | 0.850 |
| Modular neural network | 0.147 | 0.668 | 0.383 | 0.932 | 0.868 |

The results from this experiment show that the HMMMs framework clearly outperformed both neural network models.

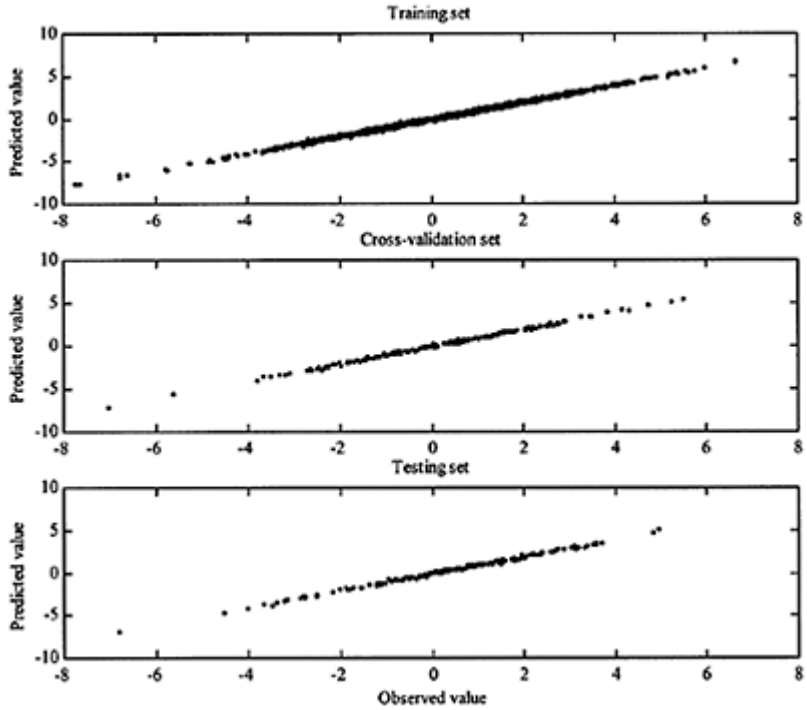



Figure 5.7. Scatter plots of the measured and predicted values for the three data sets.

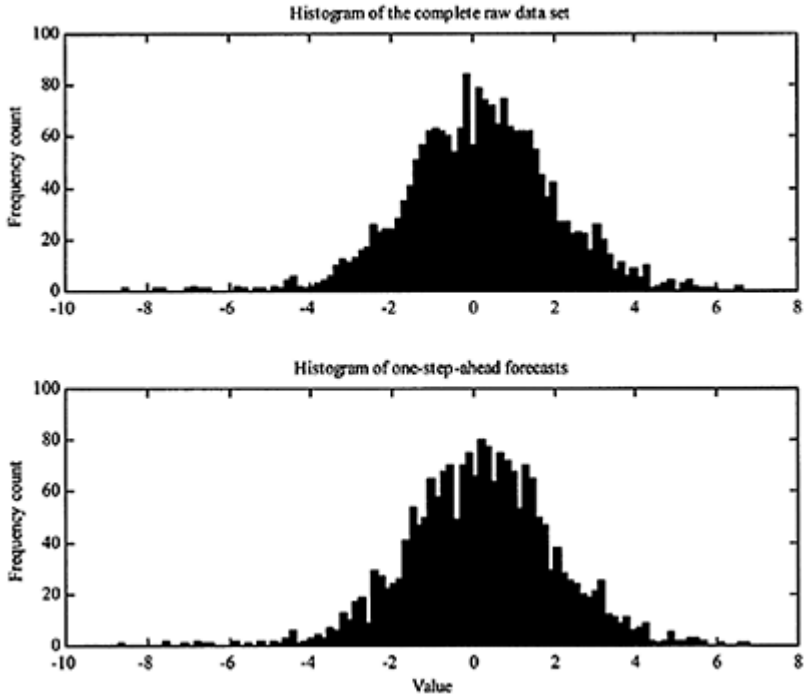



Figure 5.8. Histogram of the complete data set with the forecasts.

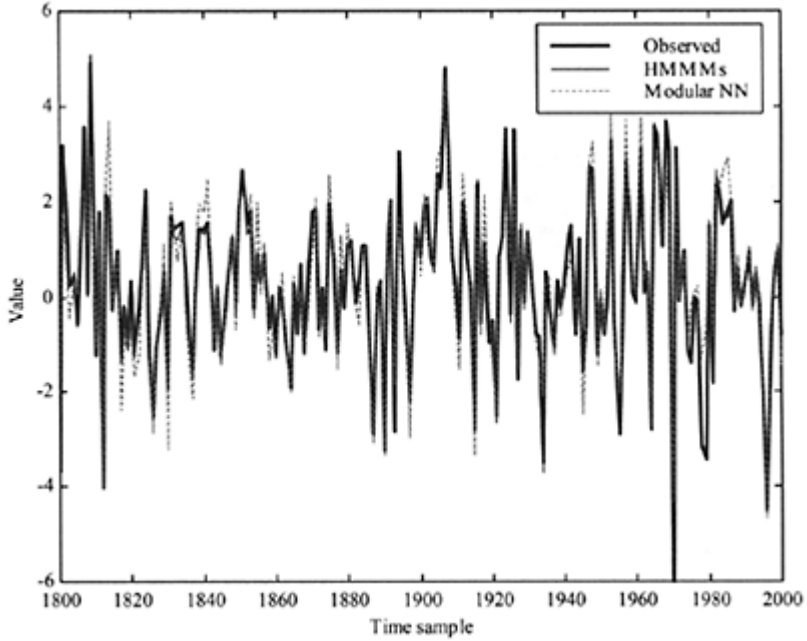



Figure 5.9. Observed and predicted times series (testing set).

*5.3.2 Different autoregressive processes*

The time series for the second experiment were generated using two different autoregressive experts using the following relation:

$$x_t = \mu + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \varepsilon \tag{5.35}$$

where $\phi_1$ and $\phi_2$ are the autoregressive model parameters, μ is the constant (intercept), and ε is the random (noise) component. The parameters $\phi_1$ and $\phi_2$ were varied in such a way that the two experts $M_1$ and $M_2$ will distinguish and satisfy the following conditions (Box and Jenkins, 1970):

$$\phi_1 + \phi_2 < 1; \quad \phi_1 - \phi_2 < 1; \quad -1 < \phi_1 < 1 . \tag{5.36}$$

The following values were used for the two experts:

$M_1$: $\phi_1 = -0.15$, $\phi_2 = 0.75$, $\mu = 112.0$ and 10% of additive zero-mean noise ε.

$M_2$: $\phi_1 = 0.45$, $\phi_2 = -0.10$, $\mu = 112.0$ and 5% of additive zero-mean noise ε with initial values of $x_t$−1=120 and $x_{t-2}$=100.

We then generated a sequence of length 1000 of the (eventually hidden) states. The sequence was generated in such a way that for every 50 steps each of the autoregressive models was responsible for generating the observations. More precisely, the time series of the underlying dynamics consists of two different regimes, generated by the two experts $M_1$ and $M_2$. The phase space was reconstructed using time lag τ=1 and embedding dimension of $m$=5. The input data set was divided in three subsets: training (800 samples), cross-validation and testing set (100 samples each). Local linear experts with different τ, $m$ and $k$ were used to recognise the sub-dynamics. Several runs showed that the global underlying dynamics of the system is best uncovered if one uses three experts. The activation of the three experts at the different dynamic regimes is presented in Figure 5.10.

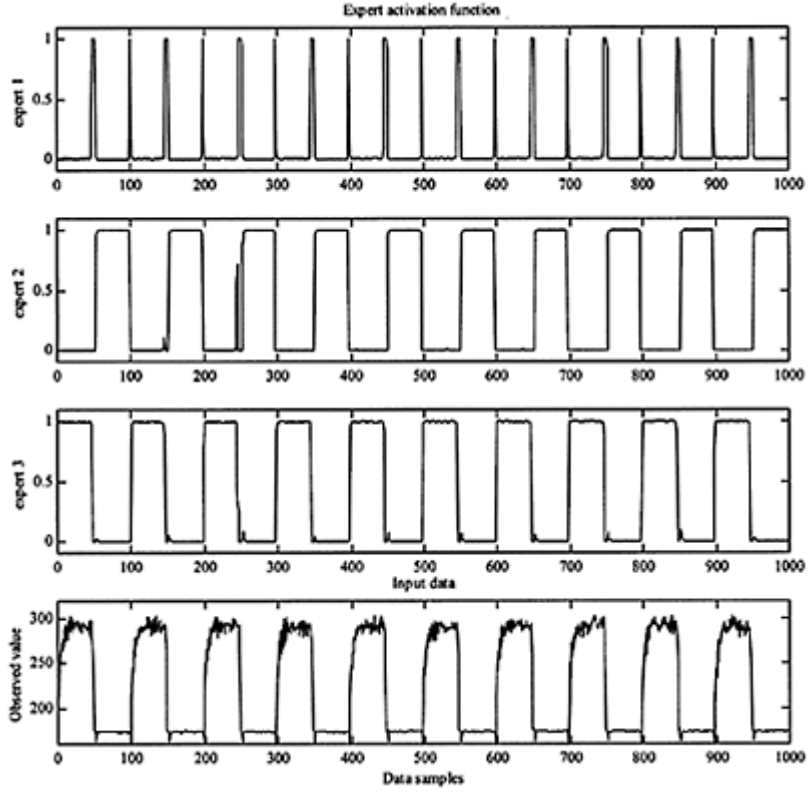



Figure 5.10. Activation of the three experts showing “specialisation “in different dynamic regimes.

It is evident that the autoregressive process (higher part of the time series) is captured by the expert 3, whereas the lower par (dynamic regime) is captured by expert 2. Furthermore, the expert 1 "specialises" on capturing the transitional part of the time series and acts as a "garbage collector". The gating process between the dynamical regimes is described by the HM model, whose model parameters (transitional and emission probabilities) are estimated using the forward-backward and EM algorithms described previously. The transitional probabilities of staying in the same dynamic regimes with the average time in regimes for the experts during the training cycles are presented in Figure 5.11. The results indicate that experts 2 and 3 have probability grater than 0.95 showing strong persistence where the expert 1 is activated only few time steps during the switching between the two main experts. The predictive performances of the model for the three data sets were evaluated using the performance indicators summarised in Table 5.2, and using scatter plots and histograms as shown in Figure 5.12 and Figure 5.13. The results were further compared to the multi-layered perceptron and modular neural network, Figure 5.14, which show better performance indicators for the HMMMs framework.

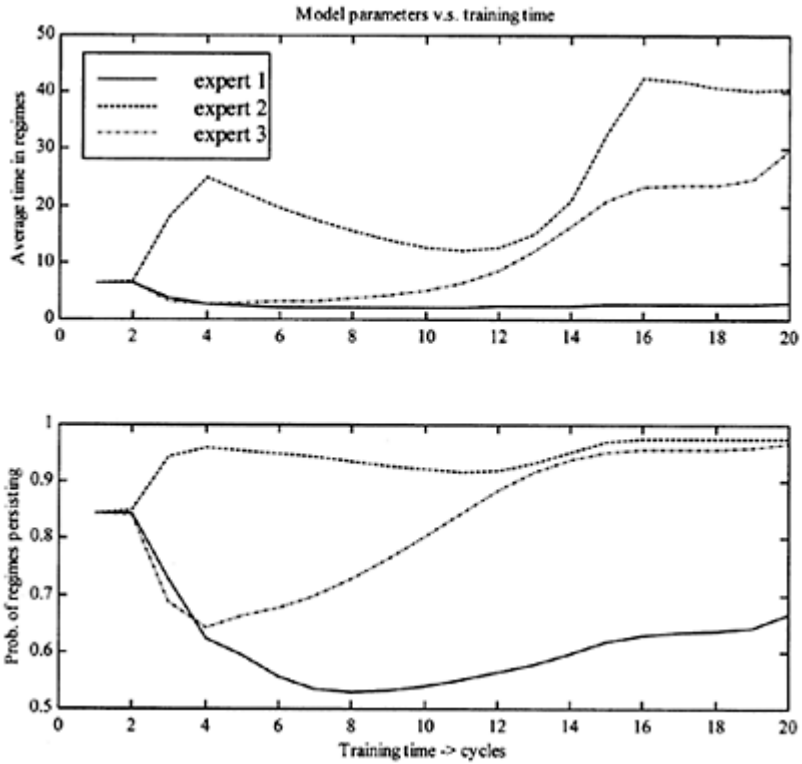



Figure 5.11. Transitional probabilities of regimes persisting for the experts.

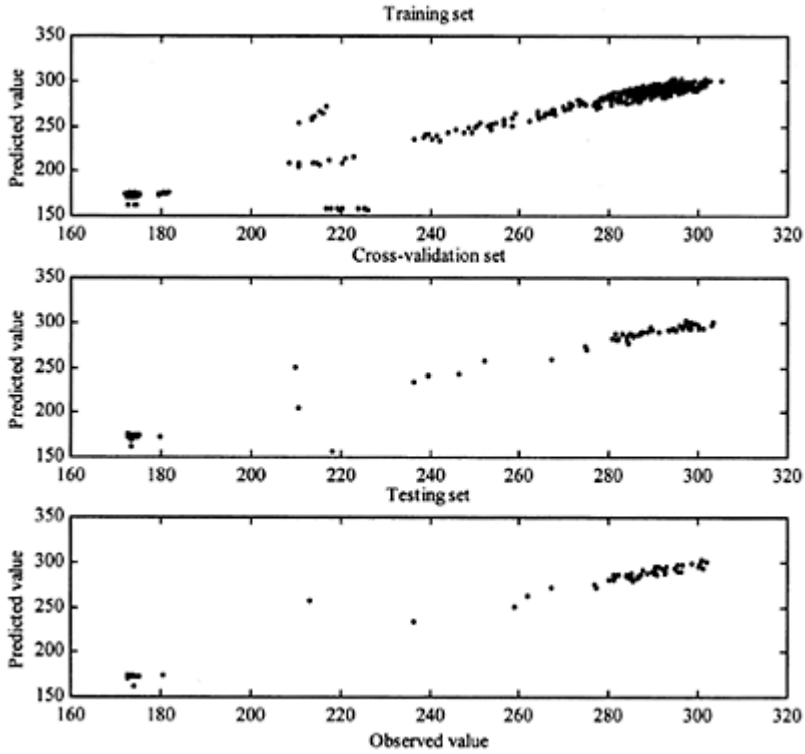



Figure 5.12. Scatter plots of the measured and predicted values for the three data sets.

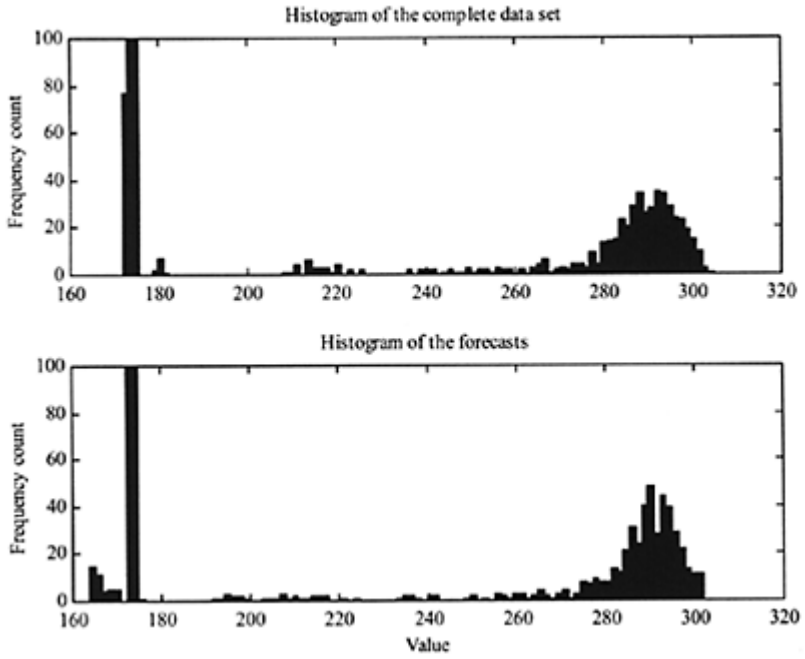



Figure 5.13. Histogram of the complete data set with the forecasts.

Table 5.2. Modelling error for the testing data set using HMMMs, MPL and modular NN.

| Model | NMSE | RMSE | NRMSE | *r* | *D* |
|---|---|---|---|---|---|
| HMMMs | 0.0087 | 5.33 | 0.0935 | 0.996 | 0.992 |
| Multi layered perceptron | 0.0214 | 8.22 | 0.1443 | 0.989 | 0.978 |
| Modular neural network | 0.0139 | 6.88 | 0.1207 | 0.991 | 0.982 |

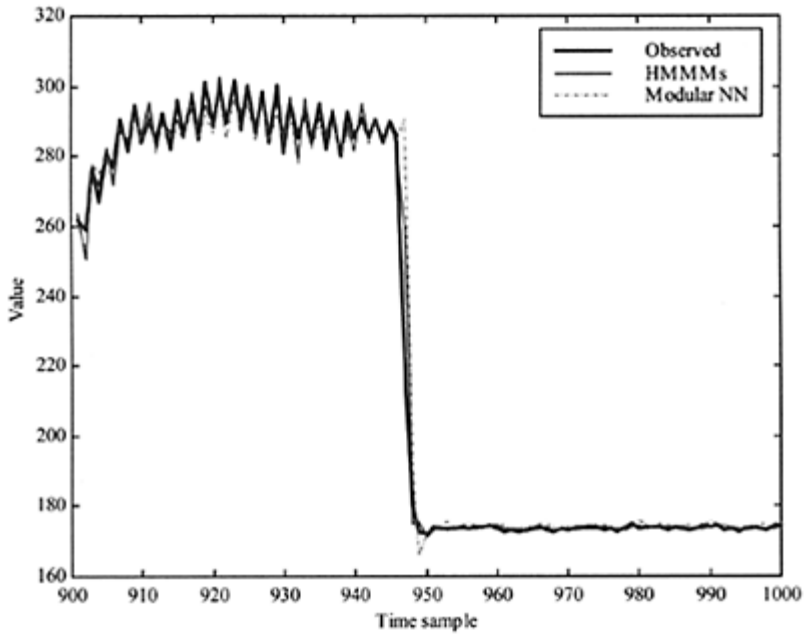



Figure 5.14. Observed and predicted times series (testing set).

*5.3.3 Mixture of stochastic processes*

In order to test the ability of the HMMMs framework to recognise the underlying dynamics of a complete stochastic process exhibiting periodic non-stationarities, time series were generated based on a normally distributed random number *(x)* with a mean $\overline{x} = 0.033$ and standard deviation σ=0.974. Two experts $M_1$ and $M_2$ with the following coefficients:

$$M_1=a_1+b_1 x;\ M_2=a_2+b_2 x$$
$$a_1=10;\ b_1=6;\ a_2=8;\ b_2=2 \qquad (5.64)$$

were used to generate one single time series consisting of two different dynamic regimes mixed up alternately at each 50 samples. The total sequence of length 1000 was divided in three subsets: training (800 samples), cross-validation and testing set (100 samples each). Two local linear experts modelling different sub-dynamics in the reconstructed phase-space were used as recognition models. The model parameter space was restricted to combination of different time delays τ=1÷4, embedding dimensions *m*=2÷12 and number of nearest neighbours *k*=1÷30. Thus, both the model parameters and the gating model were learned during the training process.

Figure 5.15 shows the expert's activation functions for the whole time series. Both experts (local linear models in this case) are capable of uncovering the the existing sub-dynamics.

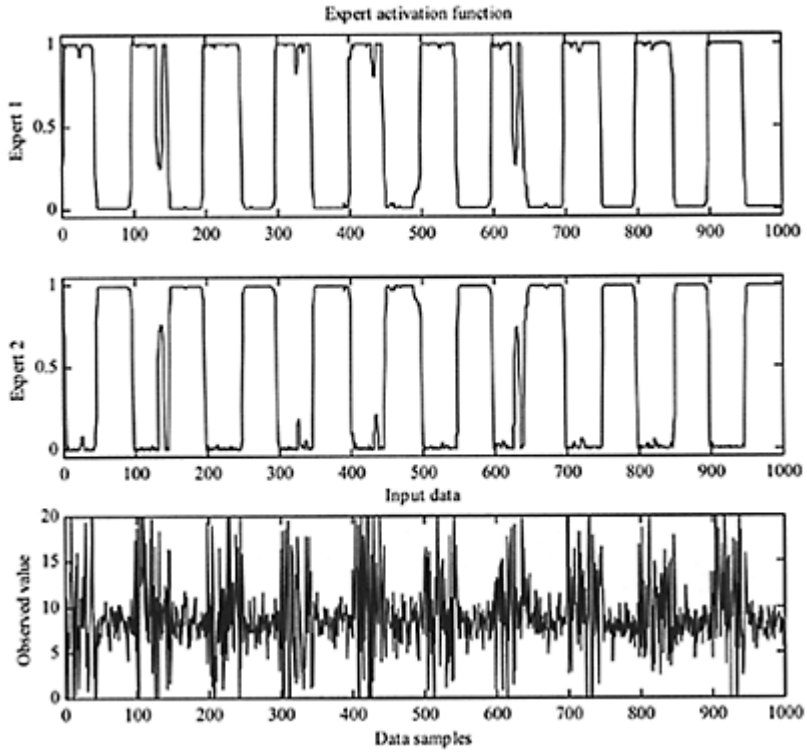



Figure 5.15. Activation of the experts capable of uncovering the different dynamic regimes.

The best performances were achieved when both experts are characterised by higher embedding dimensions ($m$>10) and low time delay $\tau$=1, indicating that the dynamics of the system does not exhibit any memory (there is no underlying structure) and tending to span infinitely the phase-space, which means completely stochastic process. Although the HMMMs framework is capable of identifying the existence of different dynamic regimes correctly, the predictive performance of this stochastic dynamics was poor. Figure 5.16 shows the one-step-ahead predictions for the testing data set for both HMMMs framework and a modular NN.

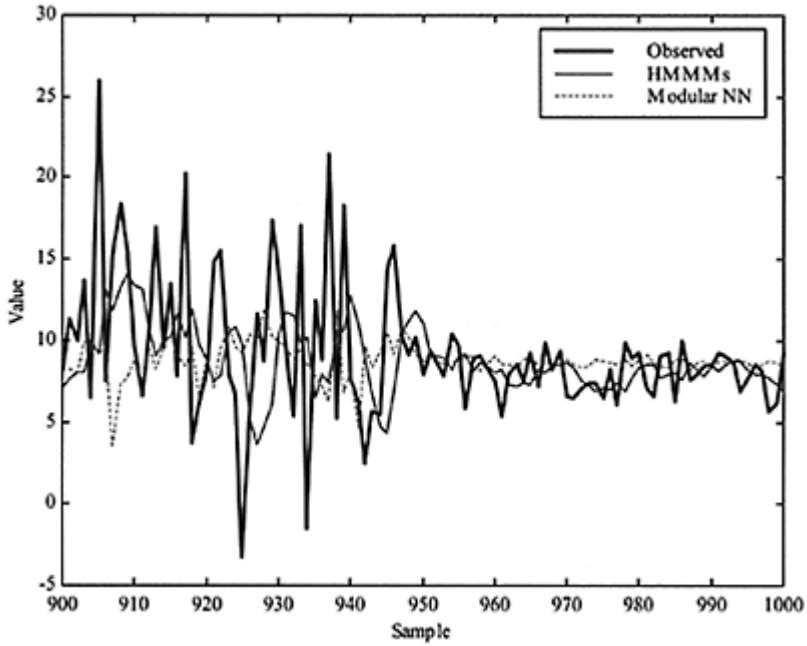



Figure 5.16. Observed and predicted times series (testing set).

### *5.3.4 Nonlinear deterministic chaos*

The last experiment consisted of testing the HMMMs framework for predicting deterministic chaotic dynamics described by time series generated by the following chaotic systems, which are usually used as benchmark problems in the neural networks and fuzzy modelling research communities:

1. Lorenz model and
2. Mackey-Glass time-delay differential equation.

#### LORENZ MODEL

Time series of the $x$ variable with a length of 10000 samples (time step $\Delta t$=0.01 sec) was generated by numerical integration of the Lorenz system as described in Example 3.1 in Chapter 3. The time delay and the embedding dimension for the reconstruction of the phase space of the Lorenz system are estimated as $\tau$=18 and $m$=3 respectively, as described in Section 3.3.8. This leads to the series of state vectors each having three components composed by the $x$ variable and its time delay:

$$\mathbf{Y}_n=\{x_n, x_{n-18}, x_{n-36}\} \tag{5.65}$$

As previously discussed and demonstrated (see Figure 3.8) the attractor of the Lorenz system is composed of two wings, whereby the evolution of the chaotic dynamics alternate between two different dynamic regimes. The recognition models for the two individual experts were local linear models with different number of neighbours $k$ as a model parameter, and the learning task was to learn this parameter for each individual expert and the parameters of the gating procedure. The local linear models were of type:

$$x_{n+20}=F_x\{x_n, x_{n-18}, x_{n-36}\} \tag{5.66}$$

where the prediction horizon was chosen to be 20 time steps ahead in order to be compatible and comparable with the multivariate experiment discussed in Section 3.3.8. The input data set was divided in three subsets: training (8000 samples), cross-validation and testing set (1000 samples each). The activation function of the two experts specialising at different dynamic regimes is presented in Figure 5.17.

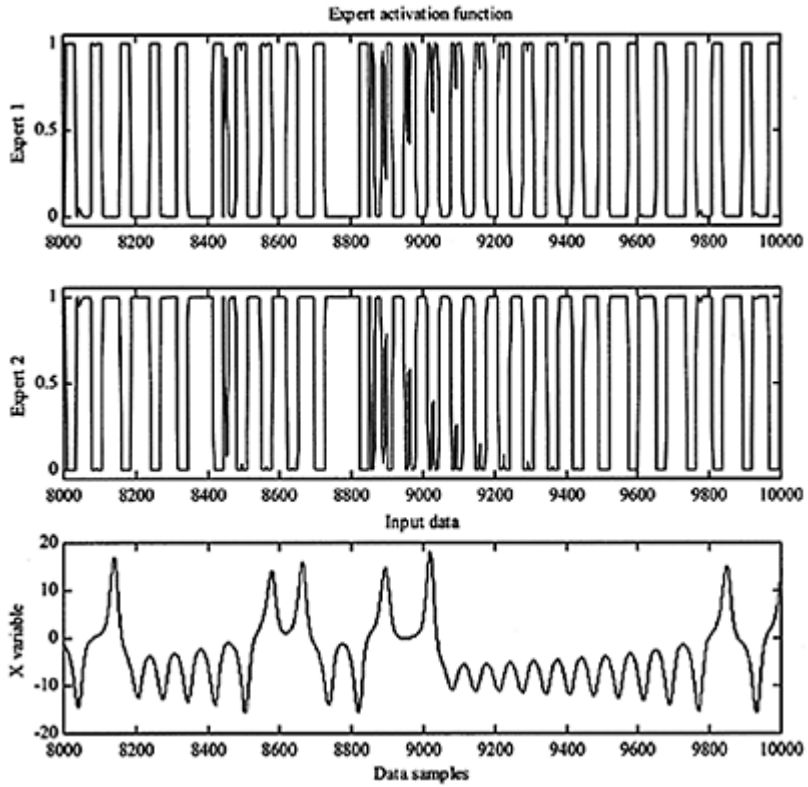



Figure 5.17. Activation of the experts capable of uncovering the different dynamic regimes of the Lorenz attractor.

The transitional probabilities of staying in the same dynamic regimes with the average time in regimes for the experts during the training cycles are presented in Figure 5.18. The results indicate that both experts have probability grater than 0.95 showing strong specialisation and persistence. Expert 1 (with optimum $k$=5) specialise on the positive wing and using negative gradients in the phase space whereas expert 2 (with optimum $k$=11) specialises on the negative wing using positive gradients for the local models in the reconstructed phase space. The gating process between the dynamical regimes is described by the HM model.

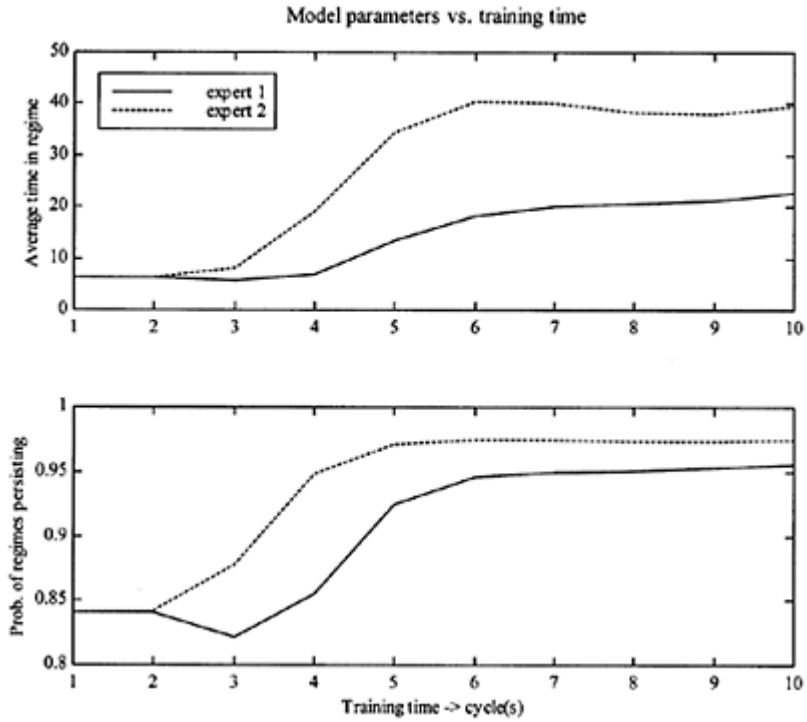



Figure 5.18. Transitional probabilities of regimes persisting for the two experts.

The predictive performances of the model for the testing data set were evaluated using the performance indicators summarised in Table 5.3, and using scatter plot and histograms as shown in Figure 5.19.

Table 5.3. Modelling error for the testing data set using HMMMs, modular NN and FIS.

| Model | NMSE | RMSE | NRMSE | *r* | *D* |
|---|---|---|---|---|---|
| HMMMs | 0.00107 | 0.256 | 0.0372 | 0.998 | |
| Modular neural network | 0.00154 | 0.376 | 0.0546 | 0.997 | |
| Fuzzy inference system (FIS) | 0.00161 | 0.424 | 0.0616 | 0.996 | |

The model predictions were further compared to modular neural network, and to fuzzy inference system (FIS) incorporated in the MatLab simulation environment and described by Jang (1991, 1993). The result in Table 5.3 and Figure 5.20 show better performance indicators for the HMMMs framework. Compared to the multivariate local modelling (see Section 3.3.8) the HMMMs framework shows slightly better results. However in this case the phase space was reconstructed using only the x variable of the Lorenz system. It ti further worth mentioning that for one step-ahead prediction, the HMMMs framework gives exact match between the observed and predicted time series.

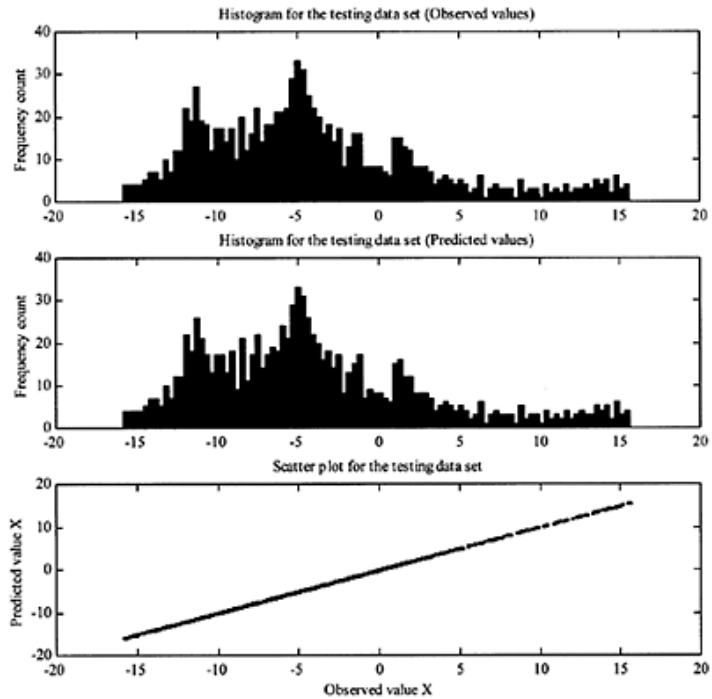



Figure 5.19. Histograms with scatter plot of the observed and predicted values of the variable X for the testing data set.

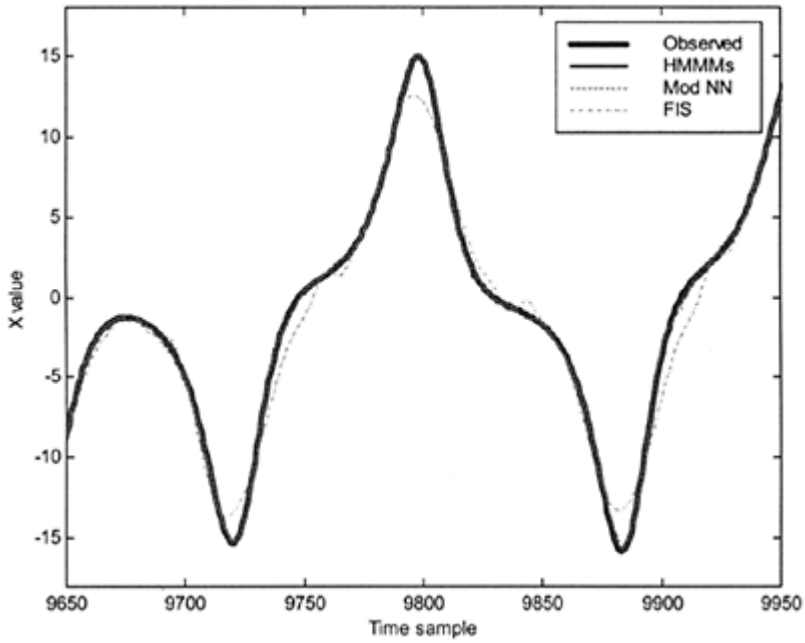



Figure 5.20. Observed and predicted times series (testing set). Part of the testing set is presented in this figure in order to compare different models.

MACKEY-GLASS MODEL

A number of models concerning biological and meteorological processes is given in terms of nonlinear delay-differential equations. Many examples exist in ecology for describing certain population dynamics (Scheffer and Kot, 1989) and other models of the same class appear in the representation of different biological oscillators (Glass and Mackey, 1988). A well-known example of these is the Mackey-Glass equation for the control of white blood cell production (Mackey and L.Glass. 1977). The Mackey-Glass model is given by the following equation:

$$\frac{dx}{dt} = F\left(x_{t-\delta}\right) - Bx_t \tag{5.67}$$

where $x$ is the density of the circulating white blood cells (WBC), $B$ is the random WBC destruction rate and the function $F$ is the current flux of new WBC into the blood in response to the demand created at a time $\delta$ in the past. The particular form of the flux function $F$ chosen by Mackey and Glass in the model results in the following form of the nonlinear delay differential equation:

$$\frac{dx}{dt} = \frac{0.2x_{t-\delta}}{1+x^{10}_{t-\delta}} - 0.1x_t \tag{5.68}$$

The dynamical evolution of this model shows very rich and complex dynamics having an infinite-dimensional state. For the delay parameter $\delta$>16.7 the dynamics of the system is chaotic. Since the Mackey-Glass chaotic dynamics has been extensively examined by researchers as a significant benchmark to compare different data modelling approaches, the above results can be useful also in the context of testing the HMMMs framework. The time series for the variable $x$ was generated by numerical integration using the fourth-order Runge-Kutta method for initial condition $x(0)$=1.2, delay parameter $\delta$=17, and $x(t)$=0 for $t$<0. The time step used in the numerical integration was $\Delta t$=0.1 sec with 10000 samples in total (t=1000 sec). The optimal time delay and the embedding dimensions for the reconstruction of the phase space were estimated as $\tau$=6 and m=4 respectively. The prediction horizon was set to 6 time steps ahead. The input data set was divided in three subsets: training (8000 samples), cross-validation and testing set (1000 samples each). Figure 5.21 shows the projection of the reconstructed phase space and the attractor in three dimensions.

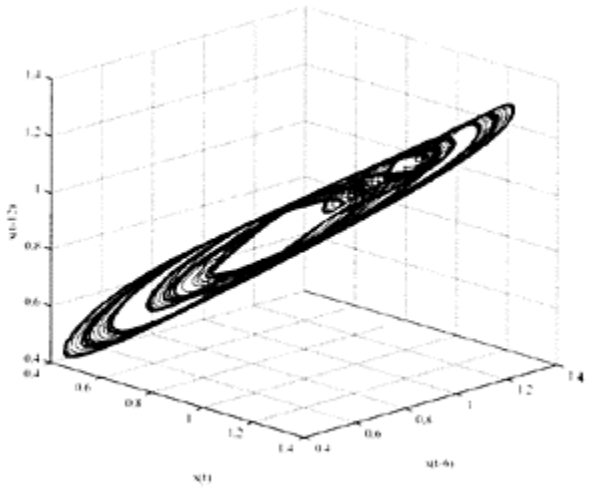


Figure 5.21. 3D view of the reconstructed attractor of the Mackay-Glass model.

Local polynomial models with parameters $k$ (number of dynamical neighbours) and $n$ (order) were used as recognition experts in the reconstructed phase space. The number of the experts, their parameters and the gating model were learned during the training process. The experiments suggested that the underlying chaotic dynamics of the Mackey-Glass model can be best uncovered using 4 local models, which specialises on different sub-areas in the reconstructed phase space. Figure 5.22 shows the activation functions of

the experts, where only expert 3 is a local liner model and the rest of the experts are local polynomials of order $n$=3 with different number of dynamical neighbours ($k$=5, $k$=8 and $k$=15 for each local expert respectively).

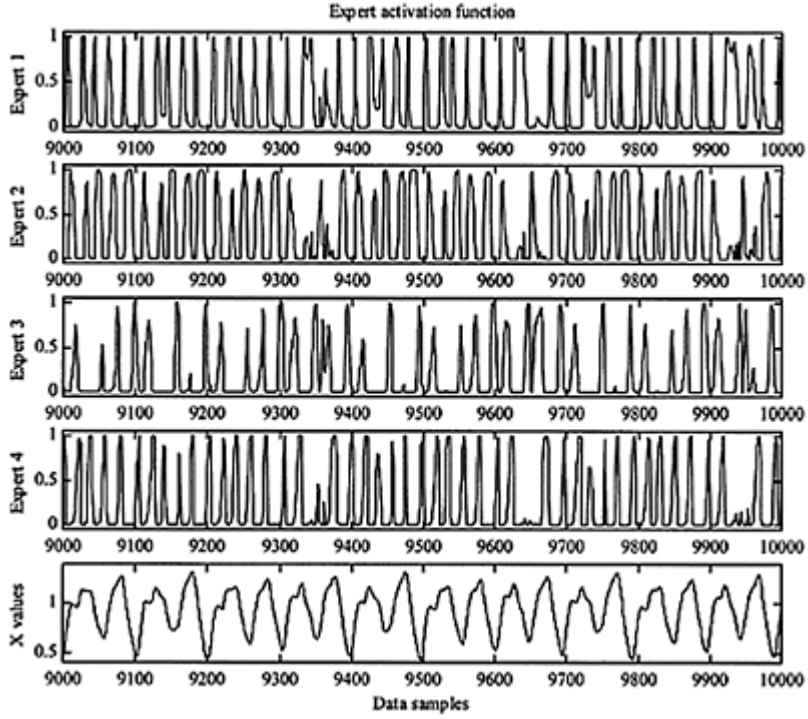



Figure 5.22. Activation of the experts (local models) for the Mackey-Glass chaotic system.

The predictive performances of the model for the testing data set were evaluated using the performance indicators summarised in Table 5.4, and using scatter plot and histograms as shown in Figure 5.23.

Table 5.4. Modelling error for the testing data set of the Mackey-Glass dynamical system using HMMMs, modular NN and FIS.

| Model | NMSE | RMSE | NRMSE | $r$ | $D$ |
|---|---|---|---|---|---|
| HMMMs | 0.000201 | 0.00315 | 0.0142 | 0.9999 | 0.9998 |
| Modular neural network | 0.000267 | 0.00398 | 0.0178 | 0.9996 | 0.9992 |
| Fuzzy inference system (FIS) | 0.002831 | 0.01134 | 0.0506 | 0.9984 | 0.9968 |

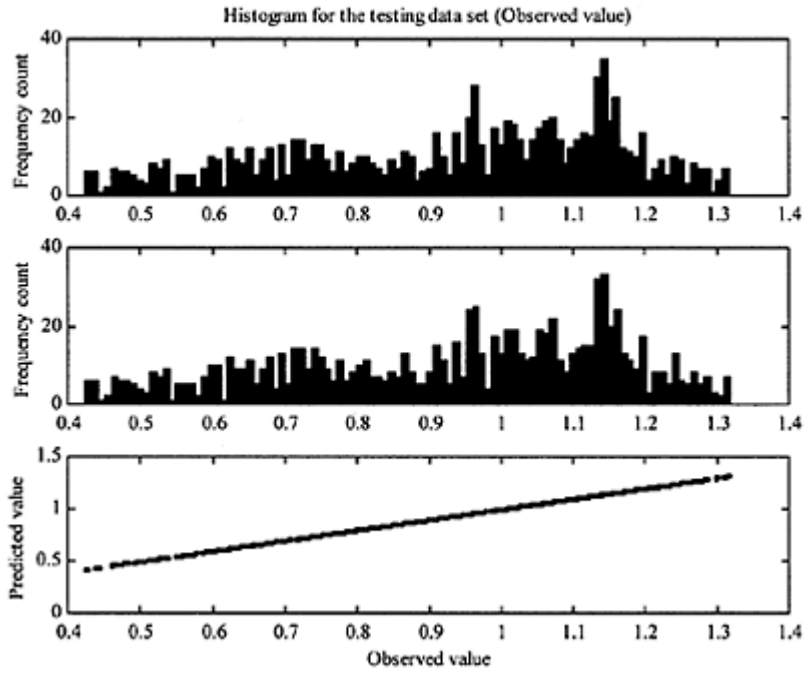



Figure 5.23. Histograms with scatter plot of the observed and predicted values of the Mackay-Glass time series for the testing data set.

As in the previous example with the Lorenz model, the difference between the original (generated) Mackey-Glass time series and the HMMMs predictions is very small. The model predictions were further compared to modular neural network, and to the fuzzy inference system (FIS) incorporated in the MatLab simulation environment. Figure 5.24 represents plot of all model predictions for a small part of the testing data set. The result in Table 5.4 and Figure 5.24 again show better performance indicators for the HMMMs framework.

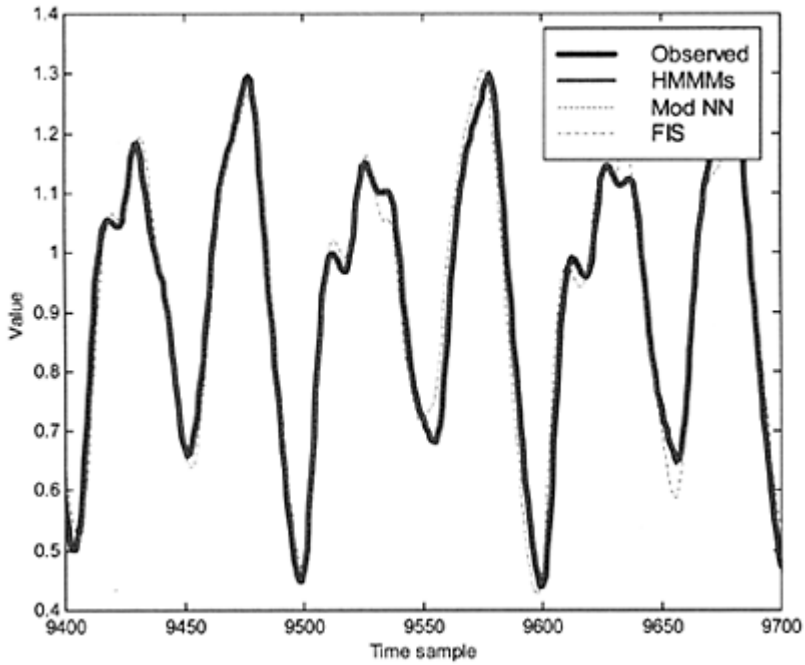



Figure 5.24. Observed and predicted times series (testing set). Part of the testing set is presented in this figure in order to compare the different models.

### 5.4 Summary

In this chapter we proposed and mathematically elaborated a hybrid modelling framework, termed as Hidden Markov Mixture of Models (experts)—HMMMs. This framework aims at separating the seemingly complex global nonlinear dynamics into couple of local subdynamics that can be modelled by separate models (experts). The separate local models through a competition specialise on modelling different parts of the reconstructed phase space of the dynamical system where the gating procedure between the models is described with a dynamic Bayesian network expressed as hidden Markov model. We have further demonstrated the wide range of dynamical systems that can be modelled by this framework on synthetic data generated by known dynamical systems. The benchmark experiments showed improved predictive performances in comparison with other nonlinear global data-driven modelling techniques, such as neural networks and fuzzy inference systems.

# Chapter 6
# Applications

## 6.1 Introduction

This chapter describes applications of the theory of nonlinear dynamics and the concept of deterministic chaos within the developed HMMMS framework for identification, modelling and prediction of hydrodynamical and hydrological systems. The selected applications cover identification and reconstruction of the underlying dynamics of several nonlinear dynamical systems: sea water level dynamics along the Dutch coast, meteorological system—precipitation and rainfall-runoff dynamics. The main objective is to demonstrate the applicability of the methods and techniques elaborated in this thesis. Most of the presented results are published in the following publications: Velickov (2003a), Velickov (2003b), Velickov (2003c), Velickov and Price (2003), Velickov (2002a), Velickov (2002b), Velickov *et al.,* (2001), Velickov and Solomatine (2000), Dibike *et al.,* (2001), Dibike et al., (2000), Solomatine *et al.,* (2001), Solomatine *et al.,* (2000).

## 6.2 Nonlinear dynamics, chaos and predictability of the water levels and surges along the Dutch coast

### *6.2.1 Introduction*

Accurate short-term operational forecasting of the surge water levels at Hoek van Holland (the entrance in the port of Rotterdam) is crucial for effective and safe ship navigation and guidance decision-making processes. The previously elaborated methods in nonlinear dynamics and chaotic time series analysis are used in this case study with the main objectives to delineate and quantify the underlying dynamics of the sea water levels, and to further assess the variability and predictability of the coastal dynamics along the Dutch coast. Phase space reconstruction, based on time-delayed embedding method, together with Poincare surface of sections and estimation of several geometric and dynamic invariants, such as dimensions, entropies and Lyapunov exponents, are used to study the sea water level dynamics. Furthermore, the shallow-water dynamic processes, which cause nonlinear interaction between the different tidal constituents and the appearance of a double low water and a distorted high water are identified and explained. Finally, multivariate local models and mixture of local models incorporating neighbouring statistics of the meteorological forcing, tide-surge interaction and the different tidal phases dynamics are elaborated and tested, showing reliable and accurate short-term forecasting performance for both total water levels and surges. In additional, an assessment of the local uncertainty and predictability of the surge dynamics is

presented and discussed. In practice, the methodology and the modelling framework presented in this case study may serve as a basis to improve the operational forecasting for ship guidance and navigation processes.

Astronomical tides generally account for about 75–80 percent of the ocean water level dynamics (water level fluctuations that occur on a time scale greater than a few minutes) in open oceans and many well-exposed coasts. Traditionally, because of the magnitude of astronomical forcing, analysis of the water levels has usually emphasised linear methods that decompose water levels into “tides” and “other” (usually meteorological) components. The amplitudes and phases of the tidal constituents, driven by the astronomical motion of the earth, moon and sun (with known periods), are then estimated using some linear methods, such as Fourier analysis, response analysis or linear regression methods. However, the water level dynamics in coastal and estuarial swallow-water areas, such as the coastal zone of the Netherlands, may differ significantly from the astronomical estimated constituents due to the nonlinear effects that include meteorological forcing, tide-surge and tide-current interactions, and tidal deformations caused by the complex topography and river discharges.

Sea and ocean water levels as complex dynamical systems are good candidates for nonlinear analysis because the governing Navier-Stokes equations including the turbulence models are inherently nonlinear. Furthermore, the sensitive dependence on the initial and/or boundary conditions of the dynamical evolution of such systems, and the broadband and continuous power spectra are one of the hallmarks of deterministic chaos, which in turn limits the predictability of such deterministic systems due to the exponential growth of small perturbations and instabilities in the system. This study analyses the underlying sea water level dynamics at seven locations along the Dutch coast: five of which originate from complex coastal locations, and two from the open sea (refer to Table 6.2.1 and Figure 6.2.1).

### *6.2.2 The data*

The water level data from several coastal stations along the Dutch coast are monitored by the Directie Nordzee (DNZ) using pressure-based water level measuring system. Water levels are sampled at 0.0167 Hz and averaged over period of 10 minutes. Table 6.2.1 lists the seven stations chosen for this study. Each time series begins at 00:00 an January 1$^{st}$ 1990 and is available until 00:00 31$^{st}$ March 1996, which results in 337249 continuous samples in total for the 10min times series data and 54768 for the hourly times series. In addition, 10min and hourly time series data of the atmospheric pressure and wind speed/direction were provided by DNZ. The average heights presented in Table 6.2.1 are estimated by averaging the individual maximum to minimum water levels. The variance of these data for both the water levels and the residuals are shown together with the maximal ranges. Furthermore, the percentage difference (mean absolute difference) between the measured water levels and the harmonic tidal estimator used in practice are shown in Table 6.2.1 (last column).

Table 6.2.1. Coastal stations along the Dutch coast used to analyse the water level data for the period between 1990–1996 (337249 samples).

| Codes | Name | Position | Water levels | | | | Surges | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Max range [cm] | Average height [cm] | Significant height [cm] | Variance [$cm^2 \times 10^3$] | Max range [cm] | Variance [$cm^2 \times 10^3$] | % diff |
| DZL | Delfzijl | N W | 547 | 293.7 | 358.3 | 12.15 | 439 | 1.429 | 42.9 |
| EPF | Euro platform | N W | 438 | 162.3 | 219.1 | 3.87 | 357 | 0.563 | 48.7 |
| HA1 | Haringvliet 10 | N W | 507 | 204.5 | 278.1 | 6.34 | 366 | 0.677 | 42.9 |
| HVH | Hoek van Holland | N W | 471 | 171.5 | 229.4 | 4.63 | 358 | 0.708 | 50.6 |
| K13 | K13 platform | N W | 468 | 156.4 | 208.8 | 2.68 | 332 | 0.773 | 46.6 |
| VLI | Vlissingen | N W | 526 | 360.8 | 414.9 | 18.4 | 405 | 0.734 | 30.5 |
| YMD | Ijmuiden | N W | 486 | 158.1 | 215.6 | 4.00 | 376 | 0.860 | 55.9 |

Based on the average wave height, a significant wave height (H1/3) was computed which gives an indication about the character of the dynamics of the sea state at the particular location. The significant wave height is the average height of the highest one-third of all waves occurring in the analysed time period (1990–1996). For example, according to the Beaufort Scale, the dynamics of the sea surface for significant heights between 2–4 (m) can be described from “Moderate waves, taking longer form, many whitecaps, some spray” to “Sea heaps up, white foam from breaking waves begins to be blown in streaks”, which corresponds to the Beaufort Scale between 5–7 and the average wind speed/forcing between 31–61 (km/h).

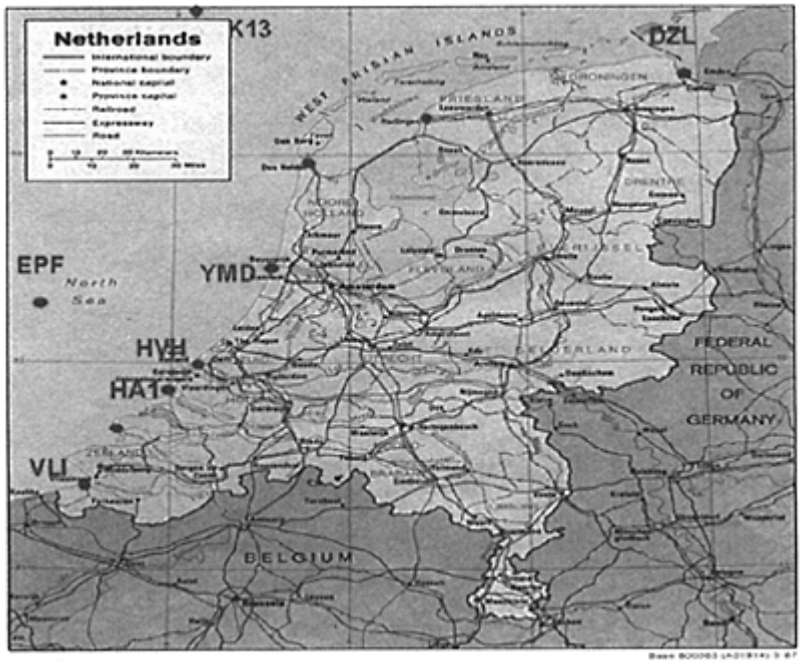



Figure 6.2.1. Location of the tidal stations along the Dutch coast.

### *6.2.3 Reconstruction of the water level dynamics from time series of observables*

The phase space of the water level dynamics was reconstructed using the methods exploring the dynamic or metric properties of the data described in Chapter 3. Figure 6.2.2 schematically represents the process of phase-space reconstruction from a historic time series, i.e. the water level in this case, where the main idea is to properly estimate the time delay $\tau$ and the embedding dimension $m$.

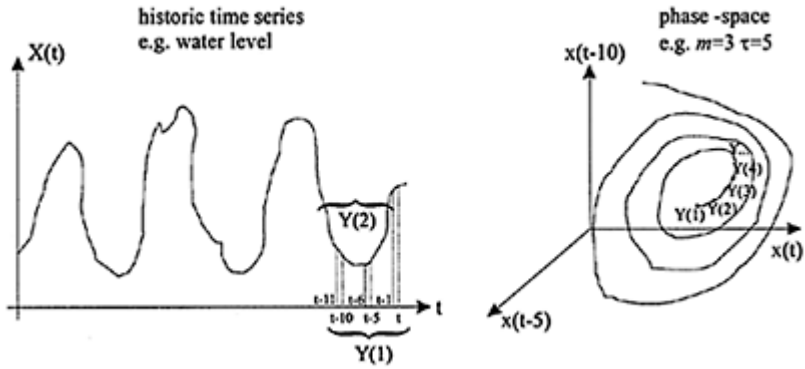



Figure 6.2.2. Schematisation of the phase-space reconstruction from historic time series. Each set of points (vector Y in equation 3.55) from the original time series is mapped to a point in the reconstructed phase space. The geometrical figure that contains the structure of the dynamical evolution of the system is called an attractor (if one exists).

Having reconstructed the phase-space of the dynamical system using the time series from the long historical records, we can further use this information to model the dynamics of the system based on *"dynamic neighbours"*. Thus, we can model the attractor in phase space, that is, find the proper local mapping functions that map the trajectory in the future, in order to predict ahead for a given time horizons (1 hour, 3 hours, 6 hours, 10 hours etc.). This can be justified by looking at "similar dynamic neighbours". These are events which happen in the past and learn from their evolution for the required prediction horizon. This local modelling process is schematically presented in Figure 6.2.3.

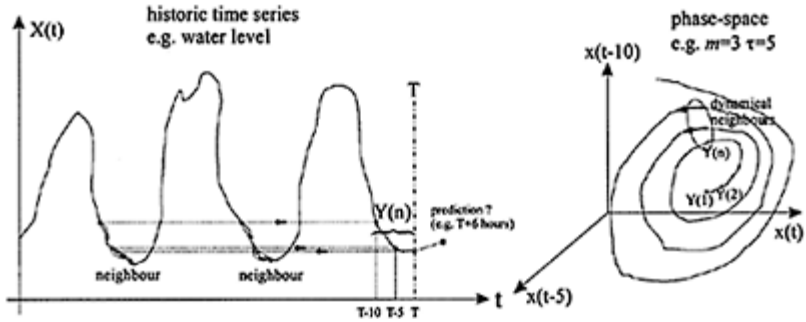



Figure 6.2.3. Illustration of the search for dynamic neighbours and their dynamical evolution in history. Based on the information of their dynamic evolution in the past one can forecasts the future evolution of the dynamics in phase space and thus the time series.

The correlation dimension $d_c$, which is used to assess the embedding dimension $m$, was estimated from the time series using the methodology described in Section 3.3.2. Figure 6.2.4 shows the correlation integral for the water level data (Hoek van Holland) at different length scales.

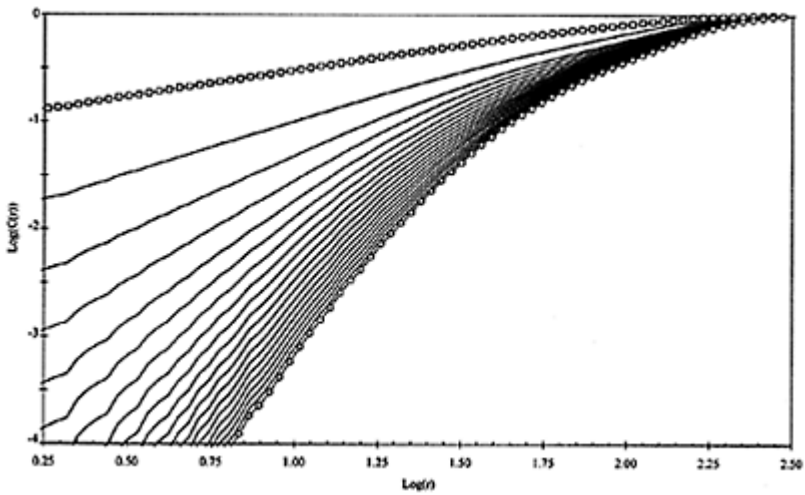



Figure 6.2.4. Correlation integral (sum) for the Hoek van Holland water level data (period 1990–1996, 10min

data). Double logarithmic plot was chosen for better visual presentation of the power law scaling between the correlation sum *C(r)* and the length scales *r*. The correlation sum was computed for different embedding dimensions (the line with squares corresponds to embedding dimension 2 and the line with open circles correspond to embedding dimension 20). After embedding dimension m=12 the lines become parallel and thus the slope (correlation exponent) saturates, next Figure 6.2.5.

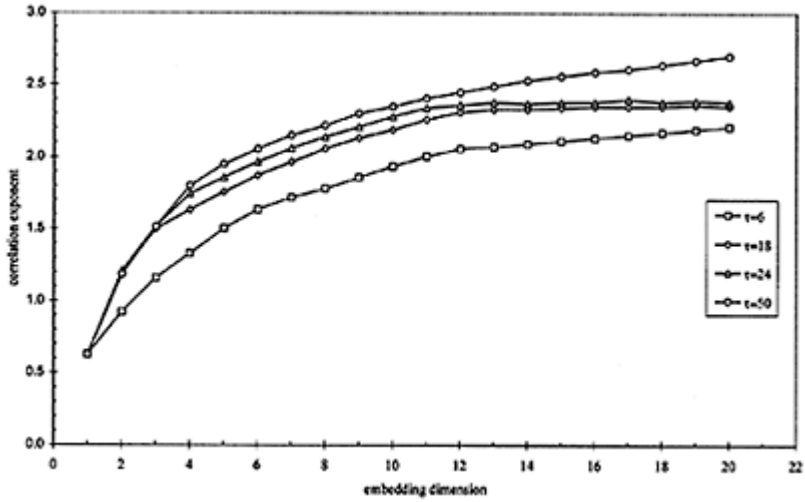



Figure 6.2.5. Relationship between the correlation exponent v and embedding dimension *m* for the Hoek van Holland 10 min interval water level data using different time delays τ. Correlation exponent increases with an increase of the embedded dimension up to a certain value and further saturates (when using time delays between τ=18

and $\tau$=24). The saturation value of the correlation exponent, that is the correlation dimension, is 2.40 (uncertainty 0.5) which indicates presence of an attractor in the dynamical system.

From Figure 6.2.5 we can find the saturation value of the correlation exponent for a properly chosen time delay for the embedding of the water level time series (the optimal time delay is $\tau$=21 in this case). This indicates the importance of finding the optimal time delay in order to correctly unfold the attractor (if one exists) in the phase space. The value of the correlation dimension of the attractor in this case is estimated to be $d_c$=2.40. Taking into account the discussion about the estimation of the embedding dimension $m$ (see Section 3.2.4), if we use the Taken's embedding theorem the embedded dimension (integer number) of the manifold which contains the attractor is $m$=6. If we use the Withney's recommendation, the embedding dimension is $m$=5. Abarbanel's recommendation (the first integer above the correlation dimension) leads us to $m$=3. The false nearest neighbours method gives an estimation of the embedding dimension as $m$=6; see Figure 6.2.6.

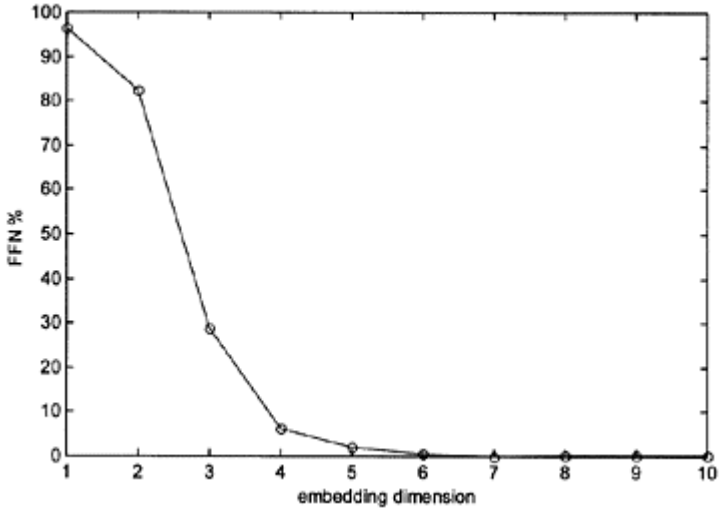



Figure 6.2.6. The percentage of the false nearest neighbours as a function of the embedding dimension for the water level data at Hoek van Holland tidal station.

This Figure shows that the percentage of the FFN drops to about 1% with the embedding dimension $m$=6, and remains unchanged for a further increase in the embedding dimension. The Lyapunov dimension estimated on the basis of the Layapunov exponents for the same data set is $d_\lambda$=5.55, (Figure 6.2.8 and Table 6.2.2), thus indicating an embedding dimension of $m$=6.

The time delay $\tau$ between successive elements in the delay vectors was estimated using the methodology described in Section 3.3.3. The autocorrelation function and the mutual information as functions of the time lags for the water level data (10min interval) at Hoek van Holland tidal station (1990–1996) are presented in Figure 6.2.7. Both functions suggest similar optimal values for the time delay of $\tau$=20 time steps, which correspond to 3.33 hours.

The Lyapunov exponents estimated from the water level time series at Hoek van Holland tidal station, using the methodology described in Section 3.3.4, are presented in Figure 6.2.8. The largest Lyapunov exponent is estimated as $\lambda_1$=0.38 (uncertainty 0.02) which indicates a loss of information of 0.38 bits/hour during the dynamical evolution of the system, and thus a loss of predictive capabilities. A theoretical assessment of the limits of predictability of the system based on the available time series indicates values between $\lambda_1^{-1}$=1/0.38=2.63 hours and $\tau/\lambda_1$=4/0.38=10.53 hours.

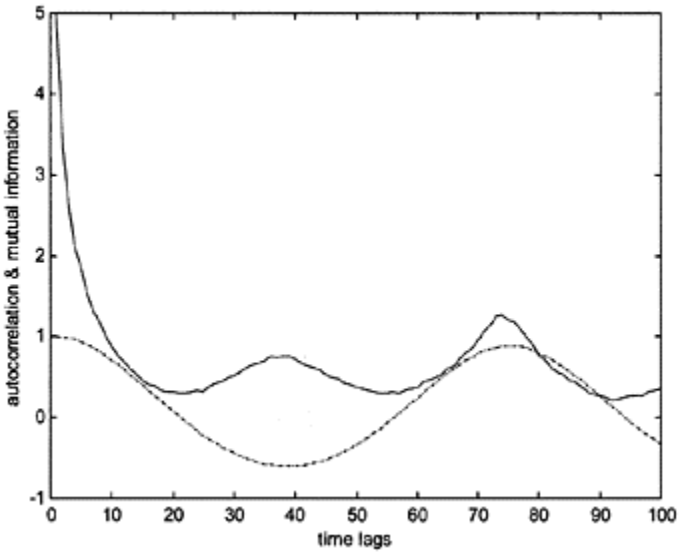



Figure 6.2.7. The autocorrelation function (dash-dotted line) and the mutual information (solid line) as a function of time lags for the hourly water level time series at Hoek van Holland tidal station.

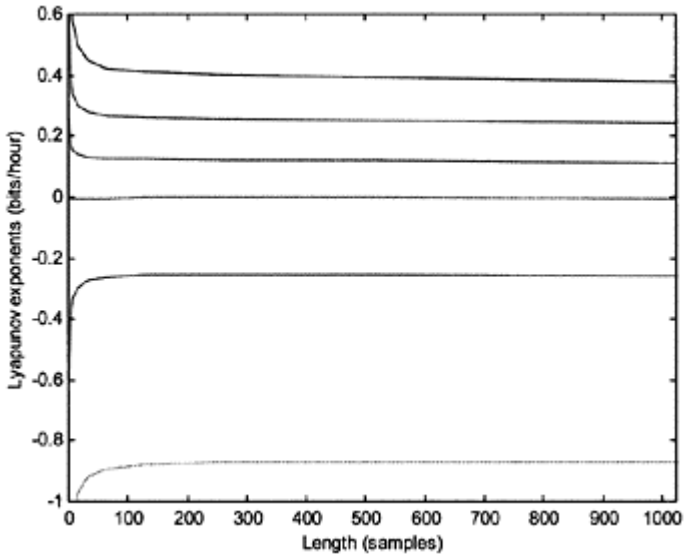



Figure 6.2.8. Estimated average local Lyapunov exponents for the hourly water level time series at Hoek van Holland tidal station in $m$=6 dimensions. The data are consistent is showing a sum of global Lyapunov exponents (the values for about 1000 steps along the attractor) that is negative.

The Lyapunov spectrum contains a large negative exponent $\lambda_6$=−0.90 which indicates the presence of strong dissipation mechanisms in the dynamics of the system. The presence of positive Lyapunov exponents and the fact that $\sum_{n=1}^{6}\lambda_n = -0.40 < 0$, provide strong evidence that the dynamics of the system is driven by deterministic chaos. Furthermore, one of the Lyaunov exponents is clearly zero, $\lambda_4$=0.0, which indicates that the deterministic motion of the system, at least in theory, can be described mathematically by a system of 6 nonlinear ordinary differential equations.

The entropy of the time series $h_2$, as an estimate of the Kolmogorov-Shinai entropy $h_{KS}$, was computed using the methodology described in Section 3.3.5. Figure 6.2.9 demonstrates the entropy estimated from the water level time series at Hoek van Holland tidal station for different embedding dimensions and time delays.

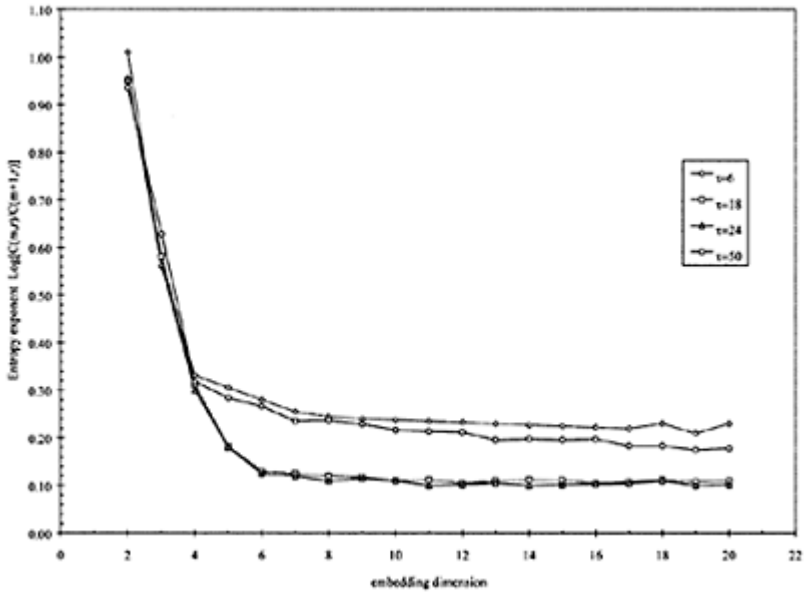



Figure 6.2.9. Dependence of the estimated entropies for the water level time series at Hoek van Holland tidal station on different embedding dimensions and time delays The results show that the embedding dimension $m$=6 and time delay $\tau$=20 provides consistent estimate of the entropy $h_2$=0.11, which is similar to the maximal Lyapunov exponent estimated from the 10min data ($\lambda_1$=0.10, see Table 6.2.2).

In Section 3.2.4 we stressed that the evolution over short time horizons can be sometimes more adequately described using the local dimension $d_l$ instead of the global embedding dimension $m$. This local dimension $d_l$ is the number of dynamic degrees of freedom that are required to model the attractor of the system and describe short-term evolutions in small regions in phase space. We could expect that this dimension is less than $m$, but for complex nonlinear systems, which may not be the case. If $d_l<m$ then the important dynamics can be captured locally with fewer degrees of freedom and the model can be simplified. If however, $d_l>m$ then the local dynamics will dictate the global behaviour of the system and the global embedding dimension (the essential degrees of freedom). To estimate the local dimension, using the same idea of FNN (see Section 3.3.2), Abarbanel and Kennel (1993) proposed a method to study the local structure of the phase space in

order to investigate if locally we require fewer dimensions than $m$ to map the evolution of the local dynamics. The main idea is for a specific number of neighbours $N_b$ and a given embedding dimension to construct local models that map the neighbours into the next time step in the same neighbourhood, but with an increased embedding dimension. When the percentage of bad predictions becomes independent of $d_l$ and is also insensitive to the number of neighbours $N_b$ it is possible to assess the correct local dynamical dimension. In the case of Hoek van Holland 10min water level data, the percentage of bad predictions (local linear), shown in Figure 6.2.10, becomes independent of the number of neighbours when the local dimension is between 5 and 6.

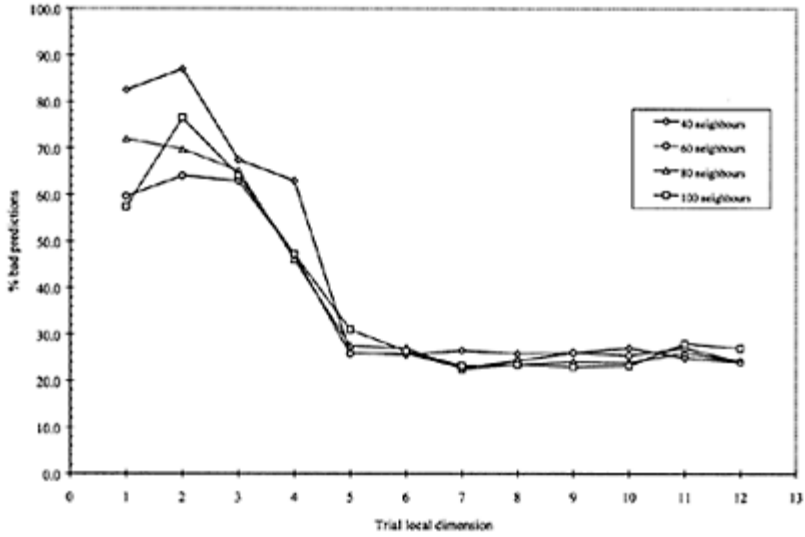



Figure 6.2.10. Local dynamic dimension for the water level time series at Hoek van Holland tidal station for different trial dimensions and number of neighbours. The results suggest local dynamic dimension of dl between 5 and 6.

### *6.2.4 Estimation of the geometrical and dynamic invariants for all tidal stations*

The same methodology and procedures were used to analyse the time series data (10 min) for the water levels and surges for all seven tidal stations along the Dutch coast. Table 6.2.2 and Table 6.2.3 summarise the results of the analysis.

Table 6.2.2. Various dimensions, time delay, entropies and Lyapunov exponents estimated using the **water level time series**

| station | Time delay [samples] $\tau$ | corr. dim $dc$ | embedding dimension $m$ | local dimension $d_l$ | KS entropy $h_{KS}$ | entropy $h_2$ | max. Lyap [bits/samp] $\lambda_1$ | Lyap. dimension $d_\lambda$ | sum Lyapunov $\Sigma\lambda_i$ |
|---|---|---|---|---|---|---|---|---|---|
| DZL | 24 | 2.53 | 6 | 6 | 0.09 | 0.10 | 0.065 | 5.27 | −0.15 |
| EPF | 19 | 2.29 | 6 (5) | 5 | 0.16 | 0.10 | 0.09 | 5.06 | −0.25 |
| HA1 | 20 | 2.41 | 6 | 6 | 0.35 | 0.19 | 0.22 | 5.26 | -0.49 |
| HVH | 23 | 2.40 | 6 | 6 | 0.19 | 0.11 | 0.10 | 5.45 | −0.38 |
| K13 | 20 | 2.25 | 6 | 6 | 0.21 | 0.13 | 0.12 | 5.32 | −0.30 |
| VLI | 20 | 2.21 | 6 (5) | 6 (5) | 0.15 | 0.10 | 0.095 | 5.02 | −0.37 |
| YMD | 18 | 2.38 | 6 | 6 | 0.19 | 0.14 | 0.13 | 5.38 | -0.42 |

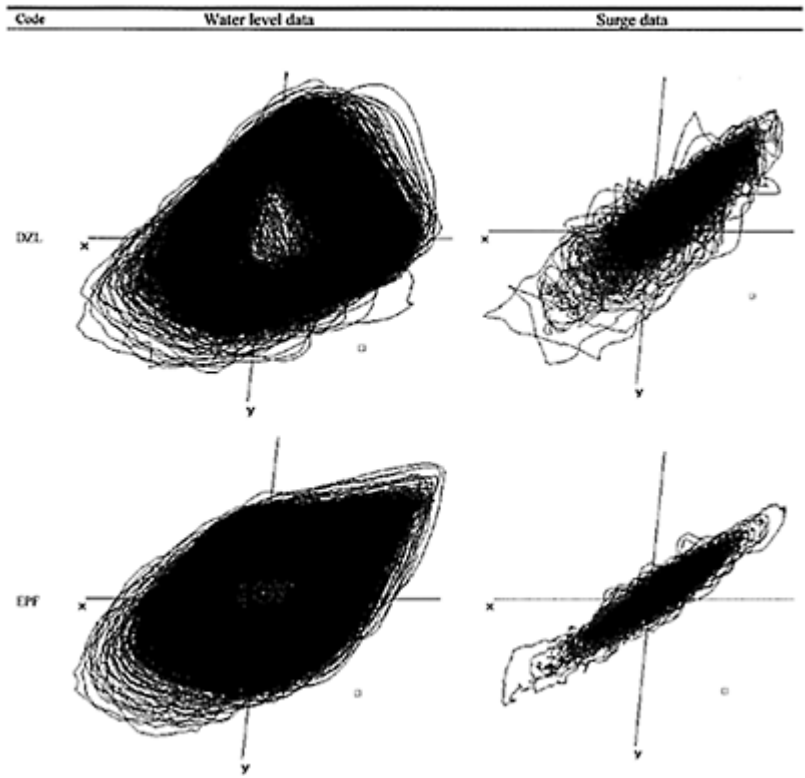
Table 6.2.3. Various dimensions, time delay, entropies and Lyapunov exponents estimated using the **residuals time series**

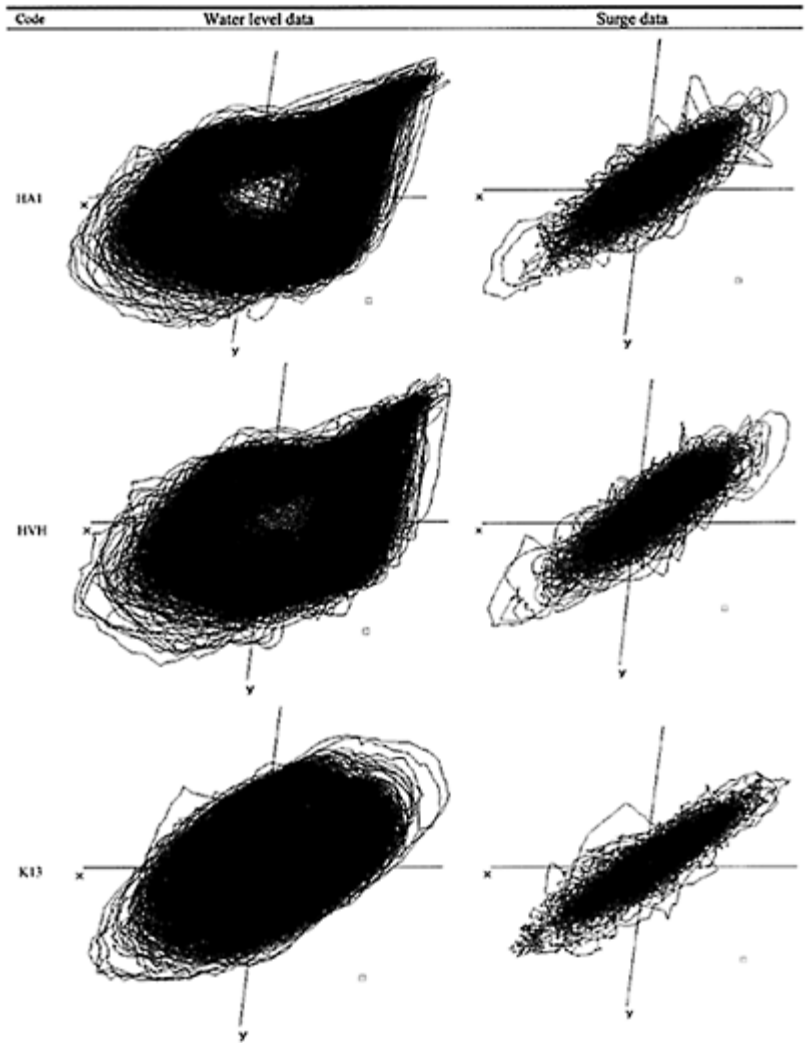
| station | time delay [samples] $\tau$ | corr. dim $d_c$ | embedding dimension $m$ | local dimension $d_l$ | KS entropy $h_{KS}$ | entropy $h_2$ | max. Lyap [bits/samp] $\lambda_1$ | Lyap. dimension $d_\lambda$ | sum Lyapunov $\Sigma\lambda_i$ |
|---|---|---|---|---|---|---|---|---|---|
| DZL | 15 | 2.87 | 6 | 6 | 1.04 | 0.66 | 0.47 | 5.89 | −0.12 |
| EPF | 10 | 1.85 | 6 (5) | 6 | 0.65 | 0.42 | 0.35 | 5.47 | −0.52 |
| HA1 | 9 | 2.96 | 6 | 6 (7) | 0.63 | 0.49 | 0.32 | 5.36 | −0.42 |
| HVH | 11 | 3.25 | 6 | 6 | 0.75 | 0.55 | 0.4 | 5.56 | −0.36 |
| K13 | 16 | 2.89 | 6 | 6 (7) | 0.68 | 0.40 | 0.36 | 5.44 | −0.50 |
| VLI | 11 | 1.92 | 6 | 6 (5) | 0.62 | 0.38 | 0.33 | 5.30 | −0.38 |
| YMD | 10 | 2.77 | 6 | 6 | 0.67 | 0.41 | 0.35 | 5.45 | −0.46 |

The results presented in Table 6.2.2 and Table 6.2.3 indicate that the embedding dimension of the total water level dynamics is not reduced due to the subtraction of the astronomical tide. On the contrary, the underlying surge dynamics reconstructed from the time series of observables at all stations are characterised by increased local dimensions and larger Lyapunov exponents and entropies, which imply shorter prediction horizons and an increased complexity. This is also illustrated by a visualisation of the reconstructed phase space. Figure 6.2.11 shows the projection of the attractors of the reconstructed phase space in three dimensions for both the water levels and the surge data (using 10min data) for all analysed tidal stations. The geometry of the attractors is summarised in Table 6.2.4.

Table 6.2.4. Geometrical characteristics of the attractors

| Code | Water levels size [cm] | mean [cm] | Surges size [cm] | Mean [cm] |
|---|---|---|---|---|
| DZL | 110.2 | 6.8 | 37.6 | −0.9 |
| EPF | 62.2 | 0.5 | 23.7 | −0.2 |
| HA1 | 76.5 | −4.2 | 26.0 | −1.9 |
| HVH | 68.0 | 6.4 | 26.6 | −1.2 |
| K13 | 51.8 | −1.5 | 27.8 | −2.2 |
| VLI | 135.8 | −2.5 | 27.1 | −0.8 |
| YMD | 63.2 | 0.2 | 29.3 | −1.2 |




Code
Water level data
Surge data
IIA1
HVH
K13
x
y

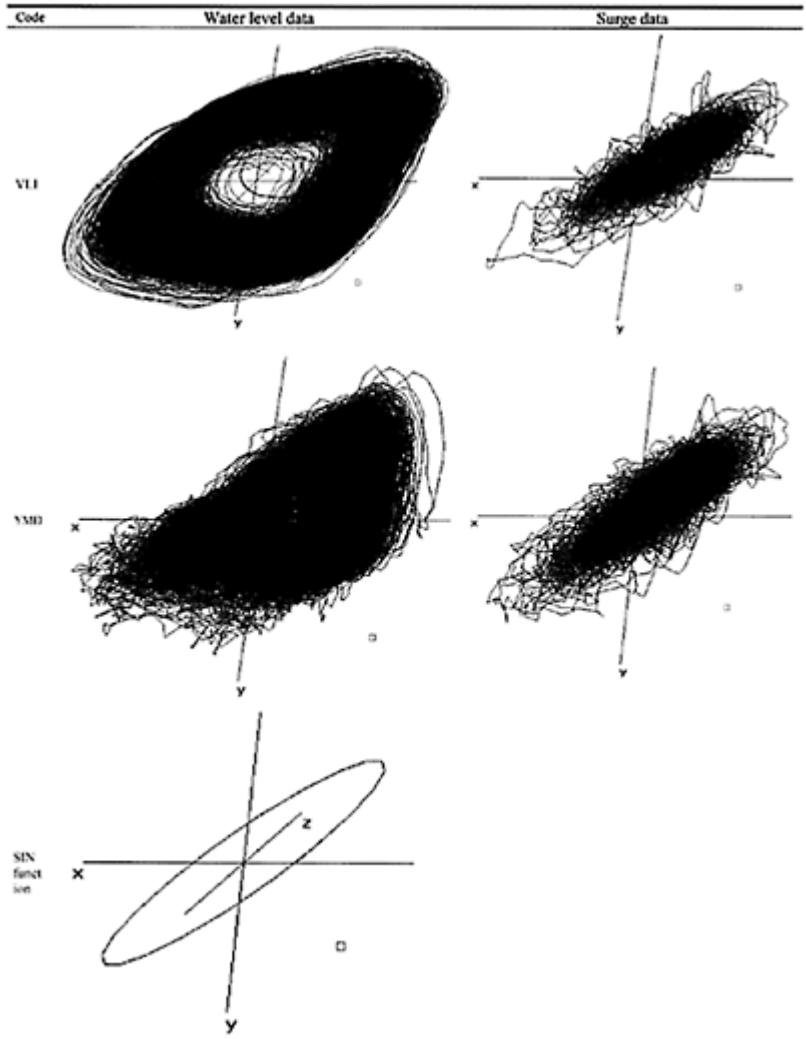



Figure 6.2.11. Projection of the attractors of the reconstructed phase space in three dimensions for both, the water levels and surge data (using 10min data) for all analysed tidal stations. The geometry of the attractors is compared with an atttractor of sinusoidal data. The attractor of Hoek van Holland data shows large distortion due to distorted high waters

and an appearance of double low waters. Similar behaviour can be noticed for the Haringfliet and the Euro Platform data.

The local modelling in the reconstructed phase space presented and elaborated in Secton 3.3.7, strongly depends on the ability to identify and select proper *dynamical neighbours* (similar events that happen in the past) and to learn the local mapping functions from their past evolutions. In order to get a sense of the neighbourhood in phase space, and to investigate the existence of similar dynamic regions of the attractor further, we can represent, for example, the ten nearest neighbours to the vector (state) Y(0) for Hoek van Holland tidal station (using embedding dimension $m$=6 and time delay $\tau$=20); see Table 6.2.5. Figure 6.2.12 represents the plot of the ten dynamic nearest neighbours to the vector at 00:10 January 1995 together with the observed and predicted (simple local model) water level time series.

Table 6.2.5. The past nearest neighbours statistics for the phase space vectors that correspond to January 1, 1995 at 00:10 (GMT+1) (time index 262944) for Hoek van Holland water level data. These are the starting points for the time series plotted in Figures 6.2.12 and 6.2.13.

| Neighbour | Time index | Date/time | Euclidean distance (cm) |
|---|---|---|---|
| 1 | 12024 | March 25, 1990, 11:50 | 14.5258 |
| **2** | **97502** | **November 9, 1991, 02:10** | **19.1572** |
| 3 | 12023 | March 25, 1990, 11:40 | 19.3391 |
| 4 | 97354 | November 8, 1991, 01:30 | 19.7231 |
| **5** | **12025** | **March 25, 1990, 12:00** | **21.4243** |
| 6 | 97353 | November 8, 1991, 01:20 | 21.7025 |
| 7 | 207499 | December 11, 1993, 23:00 | 21.7256 |
| 8 | 207500 | December 11, 1993, 23:10 | 22.2261 |
| 9 | 150856 | November 13, 1992, 14:30 | 23.7908 |
| 10 | 97503 | November 9, 1991, 02:20 | 24.8395 |

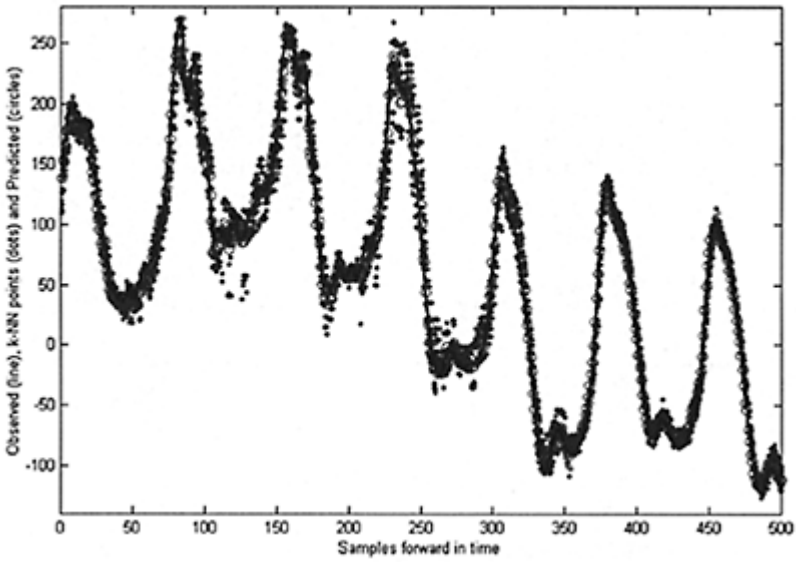



Figure 6.2.12. The 10 nearest neighbours (filled dots) to the vector at 00:10 January 1995 together with the observed water level time series (solid line) at Hoek van Holland tidal station. The zeroth local predictions (empty circles) are estimated as an average of the images of 10 neighbours.

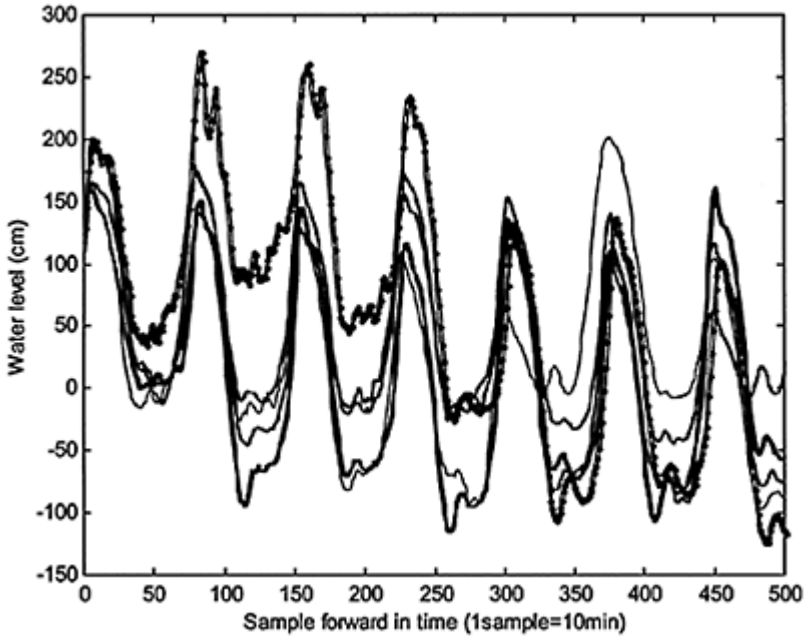



Figure 6.2.13. The dynamic evolution of the 10 nearest neighbours (solid lines) to the vector at 00:10 January 1995 (filled dots) for the water level time series at Hoek van Holland tidal station. Results show that although there are several good nearest neighbours to the initial state, they do not necessarily represent neighbours in dynamical sense in the phase space forward in time. The closest neighbours, that are the closest orbits on the trajectory in phase space, are neighbour 2 and neighbour 5 (represent with bold typeface in Table 6.2.5).

Similarly, Table 6.2.6 and Figure 6.2.14a and Figure 3.2.14b represent the statistics and the plot of the ten dynamic nearest neighbours to the vector at 00:10 January 1995 together with the observed and predicted (simple local model) surge time series.

Table 6.2.6. The past nearest neighbours statistics for the phase space vectors that correspond to January 1, 1995 at 00:10 (GMT+1) (time index 262944) for Hoek van Holland surge data. These are the starting points for the time series plotted in Figure 6.2.14 and Figure 6.2.15.

| Neighbour | Time index | Date/time | Eucledian distance (cm) |
|---|---|---|---|
| 1 | 262943 | January 1, 1995, 00:00 | 3.8730 |
| 2 | 53817 | January 9, 1995, 17:20 | 5.0990 |
| 3 | 53815 | January 9, 1995, 17:00 | 7.7460 |
| 4 | 159491 | January 12, 1993, 13:40 | 9.4340 |
| 5 | 53816 | January 9, 1995, 17:10 | 12.4097 |
| 6 | 262941 | December 31, 1994, 23:20 | 14.1067 |
| 7 | 159478 | January 12, 1993, 11:30 | 15.0000 |
| 8 | 159490 | January 12, 1993, 13:30 | 15.1327 |
| 9 | 159479 | January 12, 1993, 11:40 | 15.5242 |
| 10 | 96830 | November 4, 1991, 10:10 | 15.9687 |

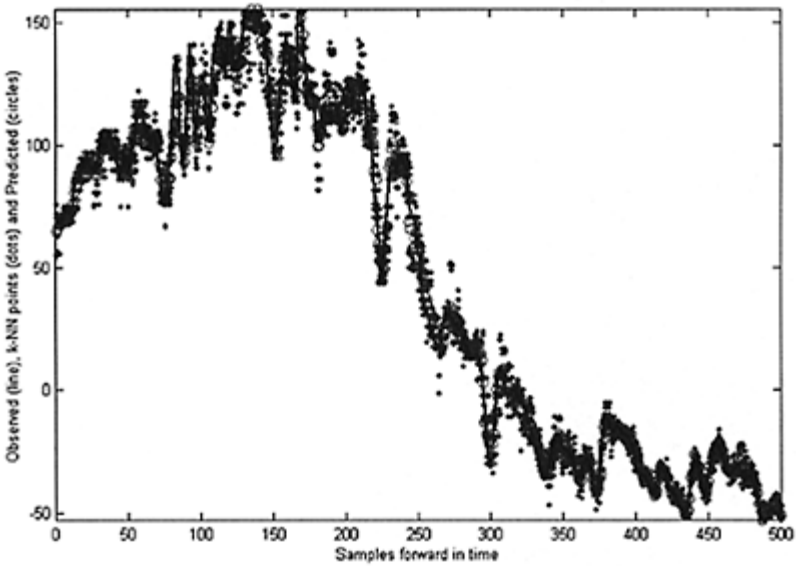



Figure 6.2.14a. The 10 nearest neighbours (filled dots) to the vector at 00:10 January 1995 together with the surge time series (solid line) at Hoek van Holland tidal station. The zeroth local predictions (empty circles) are estimated as an average of the images of 10 neighbours.

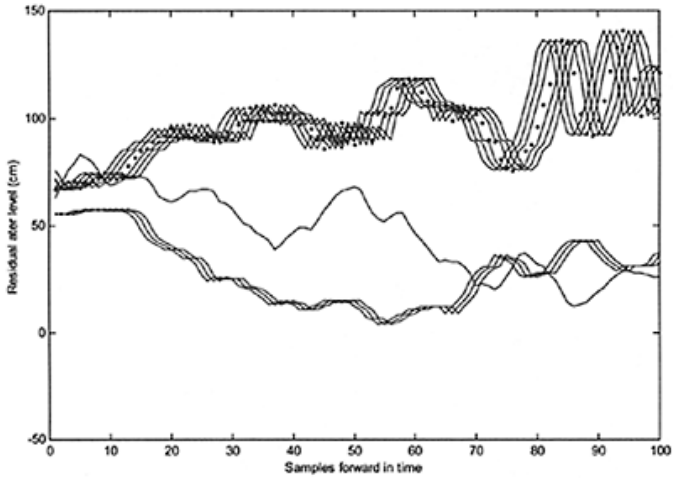



Figure 6.2.14b. The dynamic evolution of the 10 nearest neighbours (solid lines) to the vector at 00:10 January 1995 (filled dots) for the surge time series at Hoek van Holland tidal station. After approximately 20 steps forward in time, the initially nearby orbits in phase space diverge rapidly due to the presence of deterministic hyper chaos. It is also evident that some of the initial neighbours are not real "dynamical neighbours" during the evolution of the dynamics of the system.

Finally, a recurrence plot (RP) was used to show which vectors in the reconstructed space are close and far from each other. More specifically, we calculate the (Euclidean) distances between all pairs of vectors and code them as colors. Essentially, RP is a color-coded matrix, where each *[i][j]*th entry is calculated as the distance between vectors Y*(i)* and Y*(j)* in the reconstructed phase space. Thus, the recurrence plot is essentially a graphical representation of the correlation integral (Eq.3.61). The important distinction (and an advantage of the recurrence plots) is that the recurrence plots, unlike the correlation integrals, preserve the temporal dependence in the time series, in addition to the spatial dependence. In order to assess the structure of the reconstructed phase space of the surge time series, a recurrence plot was constructed and this is presented in Figure 6.2.15.

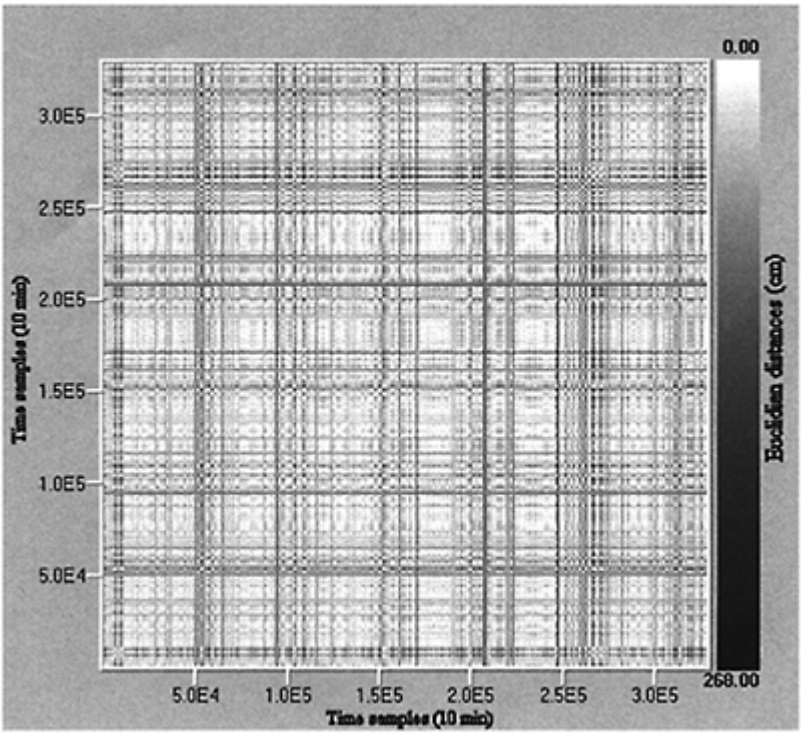



Figure 6.2.15. Recurrence plot for the surge time series based on 10min data at Hoek van Holland for the period January 1990–March 1996. The phase space was reconstructed using embedding dimension $m$=6 and time delay $\tau$=18.

The plot is symmetric along the diagonal since the distance of the ith embedded vector to the *j*th embedded vector is the same as the distance of the *j*th to the *i*th. Also, there is a diagonal line where $i=j$ (the distance between the surge vectors is 0). The recurrence plot of the surge clearly show that there is a structure in the reconstructed phase space (the lighter the color the closer the neighbours in Euclidean sense). Furthermore there is a periodic presence of vectors with higher distances especially in the winter periods, indicating that higher errors could be expect. The most variable part of the time series is year 1995 (the last part of the time series).

The results from the analysis of the total water level and surge time series data at the selected tidal stations trigger the following discussion:

Both water levels and residual data can be treated as deterministic chaotic dynamical systems that are close to the limit cycle with multiply pronounced spectral peaks.

Harmonic constituents can capture the linear periodic events, but also include long-term periodicities that are part of the true nonlinear dynamics. Furthermore several authors argue that nonlinear systems cannot be decomposed into linear subsystems in order to simplify the analysis if the true dynamics of the original system are to be retained. The results of this analysis support this finding. From Table 6.2.2 and Table 6.2.3 it is evident that the removal of the astronomical tide from the total water level dynamics did not reduce the dimensionality of the system. On the contrary, the embedding dimensions, entropies and Lyapunov exponents estimated from the residual time series show larger values, which theoretically implies shorter prediction horizons.

The general belief is that the long-term astronomical "predictions" have a greater accuracy than any other model. However, from Figures 6.2.12 and 6.2.13 we can see how the harmonic components (and any global model) might capture the average long-term behaviour, but may fail to provide accurate short-term predictions of water levels and surges (which are of great importance for everyday operational ship navigation) due to the meteorological forcing and shallow water chaotic dynamics. On the other hand, even simple zeroth order local models are able to capture the local nonlinear dynamics of the system. The lack of accuracy of the harmonic estimator to capture the complex nonlinear dynamics along the coast (see Table 6.2.1) is strong evidence that these complex dynamical systems must be analysed using a nonlinear approach.

The relatively short temporal prediction horizons described by the Lyapunov exponents and entropies (see Tables 6.2.2 and 6.2.3) stress the potential difficulties for improving any model due to the presence of chaotic dynamics. The chaotic behaviour occurs because water levels, including astronomical contributions and the contributions of many other processes, are the result of a coupled nonlinear system. The Lyapunov exponents have also significant ramifications for numerical models that are based on solutions to the hydrodynamic equations of motion. The implication of the presence of deterministic chaos in water level and surge dynamics is that estimates of future behaviour are very sensitive to mathematical formulations and assumptions, the choice of various coefficients and parameterisation, and the system's current state may be also inadequately modelled or measured. The main implication is that improvements in the forecast may require significant improvements in the accuracy of the terms, coefficients and the measurements which are used as initial and boundary conditions.

If we neglect the descriptive focus of using a numerical model to better understand the relationships between different variables and components of the underlying physical system, the main issue then becomes the temporal accuracy of moment-to-moment estimates of the time series for the water level and surges made by any model. In this respect, bearing in mind the presence of deterministic chaos in the water level and surge dynamics, local modelling in the reconstructed phase-space of the dynamical system, which uses information from the *"real" dynamical neighbours*, may give substantial forecasting improvements. Thus, the identification and selection of proper dynamical neighbours to learn from are the key issues in the local modelling approach adopted here.

### *6.2.5 Shallow-water dynamics and the dynamical neighbours*

From coastal hydrodynamics it is well-known that the amplitudes of tidal waves which are generated in the deep sea increase when they spread onto the shallow continental shelves. On the shelves the characteristics of these waves are altered by other processes including standing-wave generation and local resonances. In order to improve the selection of the dynamical neighbours for the purposes of local modelling and forecasting, it is important to identify and understand the processes that influence the local distortions which occur as the waves propagate into the shallower coastal waters along the Dutch coast. Three separate factors may contribute to the distortions: (i) Although the tidal waves still satisfy the criteria for long waves, that is, they have wavelengths which are much longer than the water depth, in shallow water the amplitudes of the waves become a significant factor of the total water depth. (ii) Secondly, the stronger currents which develop in the shallow water are resisted by the drag due to the bottom friction, a process which removes much of the tidal energy, and reduces the wave amplitudes. (iii) Thirdly, there can be a strong influence of the bathimetry and topography. The irregular coast line and varying depths impose complicated tidal current patterns. In the shallow-water areas where the currents take curved path, there must be associated surface gradients to provide the necessary cross-stream accelerations. Exact mathematical descriptions of the complicated combination of these processes are seldom possible.

The analysis of the water level time series along the Dutch coast shows that the interval from low to high water is shorter than the interval from high to low water: that is, the rise time is shorter than the fall (e.g. see Figure 6.2.16).

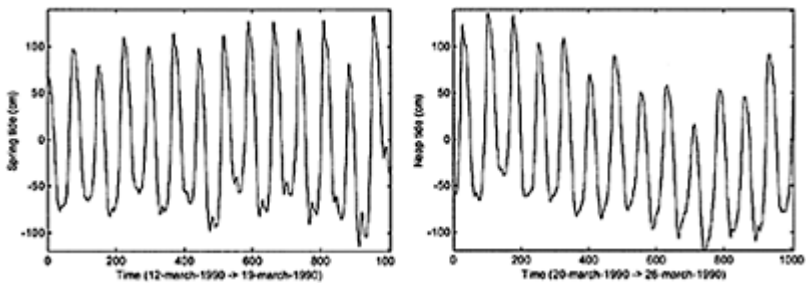



Figure 6.2.16. Typical spring and neap tide at Hoek van Holland tidal station.

In these circumstances, simple tidal predictions give times of high water which are later than the observed, and times of low water which are earlier. It is also further visible that the distortions take the form of double low and double high waters, which are more pronounced during the spring tide; see Figure 6.2.17. The possible explanation of the appearance of the double low and high water can be due to the appearance and interaction between higher shallow-water harmonic constituents, such as $M_4$, $M_6$, $M_8$ etc. acting on a short-term period of time.

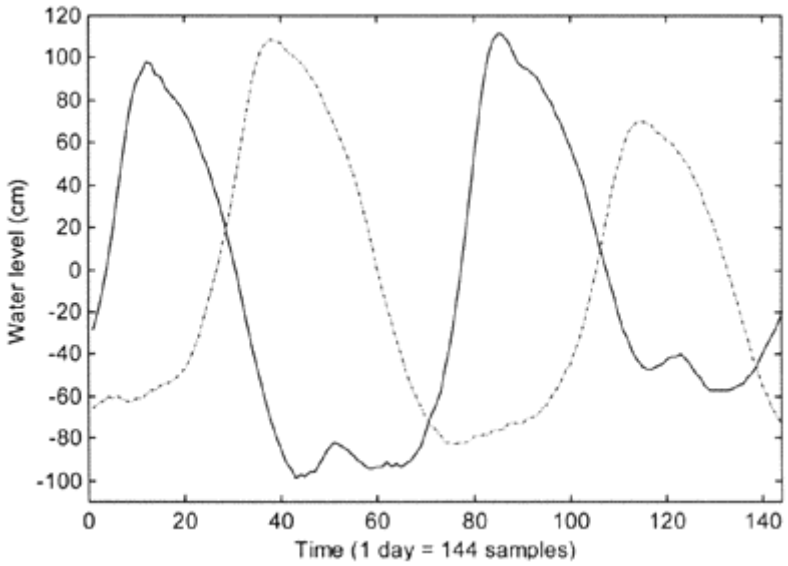



Figure 6.2.17. Spring tide (15 March 1990, solid line) and neap tide (22-March-1990, dashed-dotted line) at Hoek van Holland tidal station. Both, the double low water and the distorted peak are more pronounced for the spring tides.

The power spectra for both the water levels and surges presented in Figure 6.2.18 indicate high energy levels from these higher-order tidal components. In order to extract these constituents, as an alternative to the analysis of long period components (such as monthly and yearly constituents), we analyse the daily time series for the harmonics present in each constituent. These daily harmonics may be called $D_1$, $D_2$, $D_4$, etc. by analogue with the usual mutation of the naming of harmonic constituents.

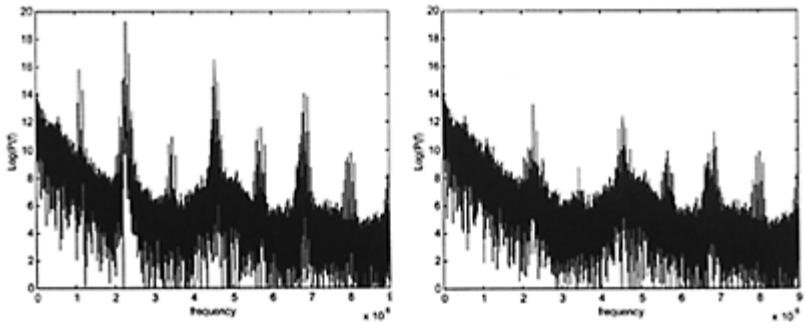



Figure 6.2.18. Power spectrum of the water levels (left figure) and the surges (right figure) time series for Hoek van Holland tidal station. Higher order harmonic components have significant contribution to the total spectral energy.

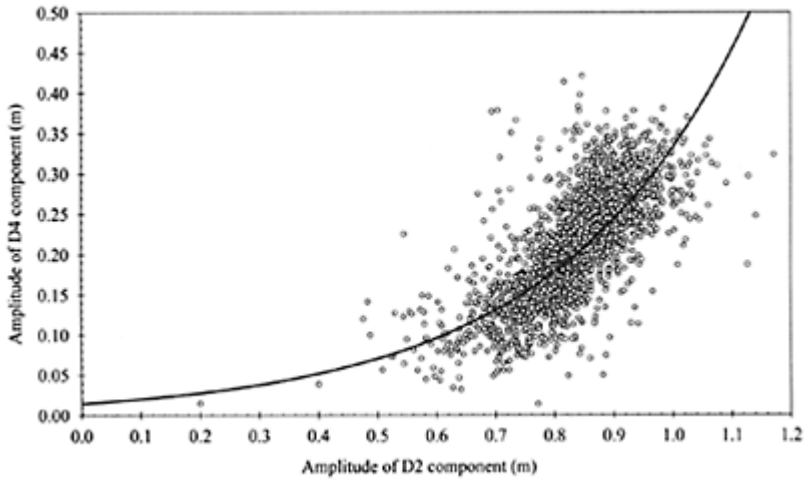



Figure 6.2.19. Scatter plot of the $D_2$ and $D_4$ amplitudes. Possible nonlinear relationship is evident.

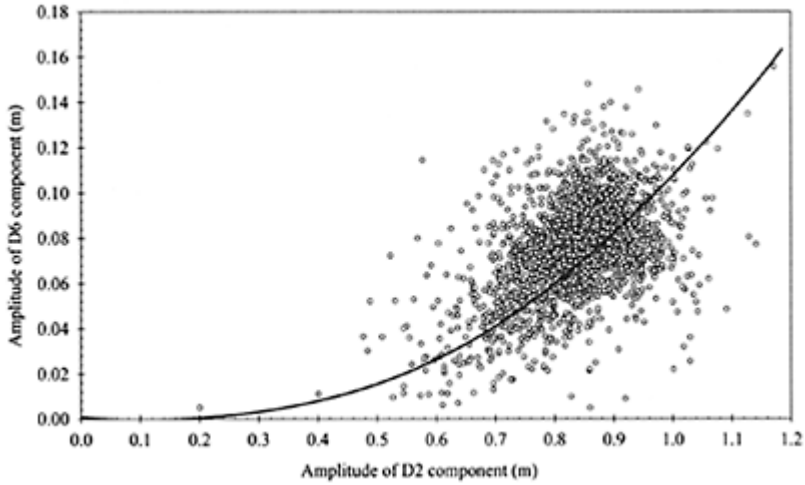



Figure 6.2.20. Scatter plot of the $D_2$ and $D_6$ amplitudes. Possible nonlinear relationship is evident.

Modulations of these $D_n$ terms are expected to take place over the spring-neap cycle in the same way as the many lunar constituents in the monthly and yearly time periods. The daily analysis gives some insight into the relationships between the constituents, which in turn gives additional information for the implementation of rules for selecting proper dynamical neighbours for the purposes of local modelling and forecasting. The results from this daily analysis are presented in Figures 6.2.19, 6.2.20 and 6.2.21.

Further to the extraction of the amplitudes for the daily $D_n$ constituents, the relationships between their phases were also analysed. Figure 6.2.22 presents the relationships between the phases of the $D_2$, $D_4$, $D_6$ and $D_8$ components respectively. It is interesting to note the appearance of a *phase-locking phenomenon* between the shallow-water daily tidal components. Animation of the scatter plot further shows that this phase-locking phenomenon occurs on different time scales between the different daily $D_n$ constituents, i.e. different regions on the graphs. A possible explanation could be the presence of standing-wave generation and local resonances in the shallow-water dynamics. It is thus very important to use this additional knowledge while searching for the "real" dynamical neighbours and learning the local mapping functions from their past evolutions for the purposes of modelling and forecasting.

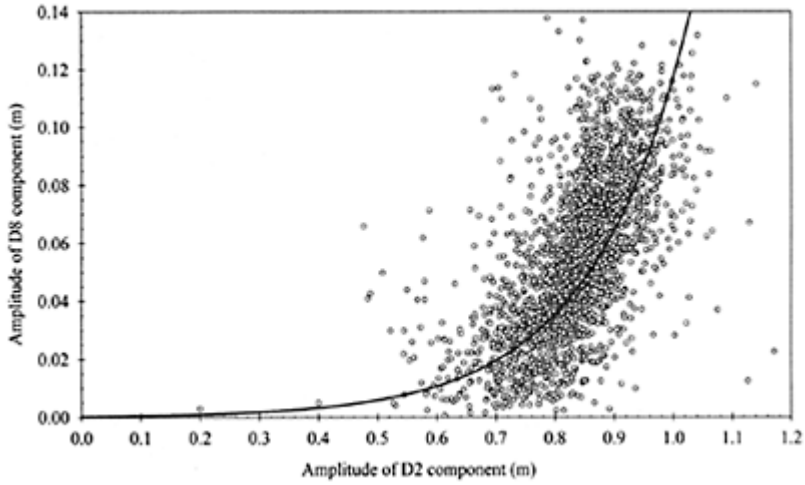



Figure 6.2.21. Scatter plot of the $D_2$ and $D_8$ amplitudes. Possible nonlinear relationship is evident.

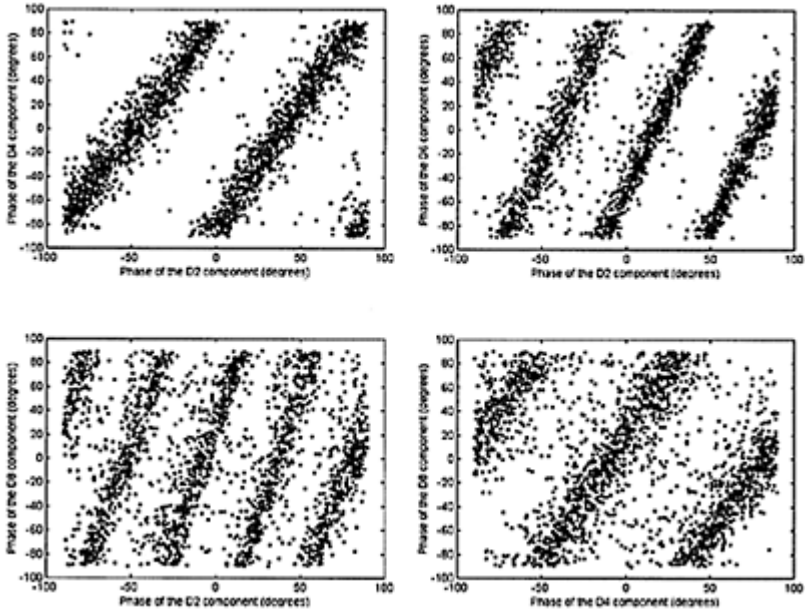



Figure 6.2.22. Phase difference between $D_2$, $D_4$, $D_6$ and $D_8$ shallow-water daily tidal components.

The phase differences between the semidiurnal component $D_2$ ($M_2$) and its subharmonics explain the double water and distorted duration of the high water during the spring tide. There are also noticeable appearances of a double high water during the neap tide. The physical explanation of the appearance of the double low water and distortion of the duration of the high waters is presented in the following Figure 6.2.23.

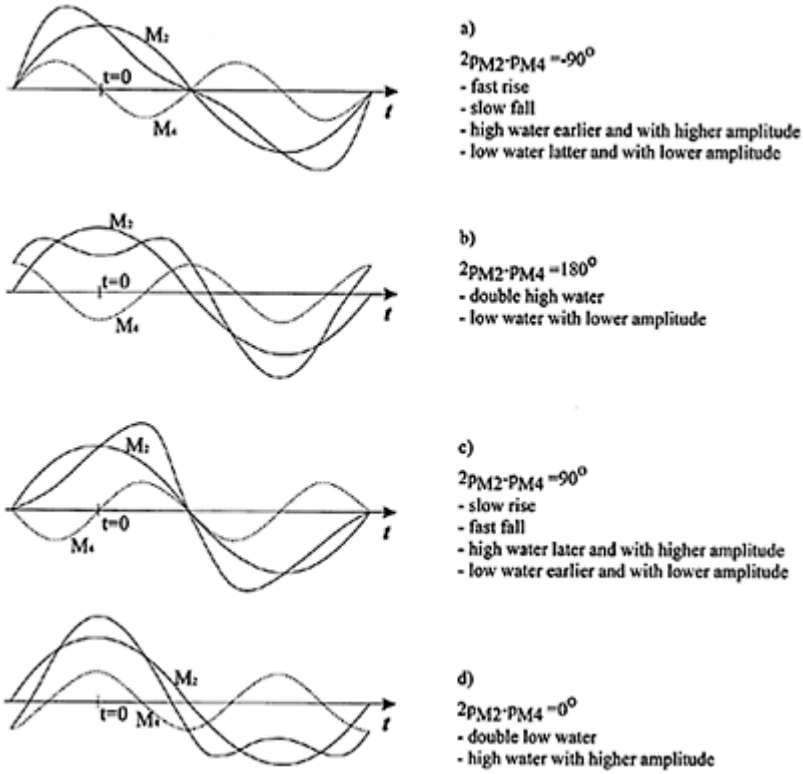



Figure 6.2.23. Schematic presentation of the shape of the composite seal-level curve, controlled by the relationship between the amplitudes and the phases of the semidiurnal $M_2$ and four-diurnal $M_4$ components in this case ($A_{M4} \approx 0.35 A_{M2}$). For swallow-water shelves along the Dutch coast, the curvature of the sea level is strongly dictated by the phase of the higher-order tidal components such as six-diurnal $M_6$ and eight-diurnal $M_8$,

whose daily amplitudes are significant in comparison to the semidiurnal component $M_2$.

### *6.2.6 Tide-Surge interaction*

The swallow-water dynamical processes, which cause interaction between different tidal constituents as already demonstrated, also cause tidal and surge components of the sea levels and currents to interact. Suppose, for example that, there is a process which depends on the square of the total sea-level:

$$\xi^2=(T+S)^2=T^2+S^2+2TS \tag{6.1}$$

then the *TS* term in this case represents the interaction between the tides and the surges. In practice this interaction is difficult to describe in terms of analytical models and some knowledge can be gained from the numerical models. An alternative method is to analyse the distribution of the positive and negative surges relative to the high and low waters from the time series of the observations, as presented in Figure 6.2.24 and Figure 6.2.25.

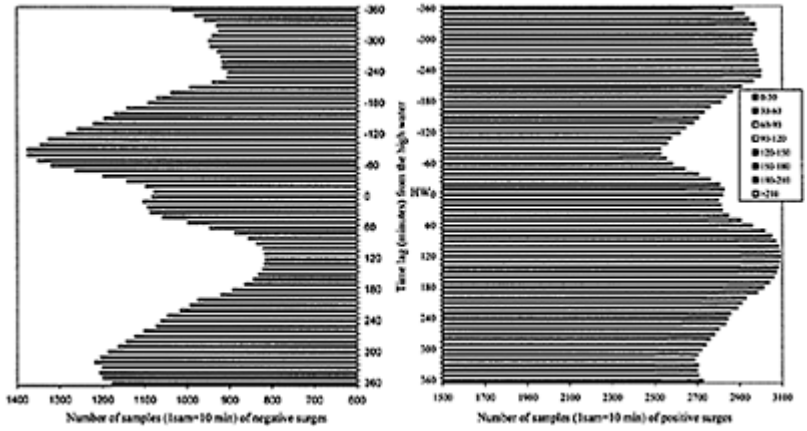



Figure 6.2.24. The distribution of the positive and negative surges at Hoek van Holland relative to the time of the high water for the period of 1990–1996, showing that the tide-surge interaction causes the classes of high surges to avoid the times (presented in minutes) of the tidal high water.

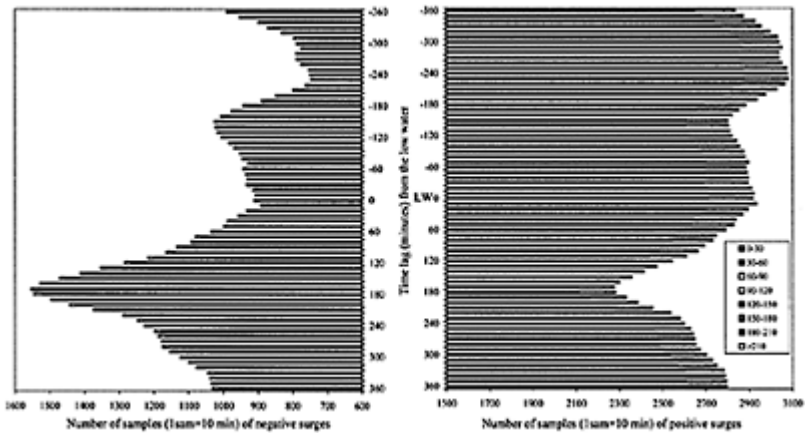



Figure 6.2.25. The distribution of the positive and negative surges at Hoek van Holland relative to the time of the low water for the period of 1990–1996, showing that the tide-surge interaction causes the classes of high surges to avoid the times (in minutes) of the tidal low water.

Tide-surge interaction on a local scale is very important because it is most apparent in shallow-water areas where large surges may be generated. This is evident from Figure 6.2.24, which shows that the pattern of interaction causes high surge peaks to avoid the time of tidal high water (HW). The probabilistic analysis on the historical 10min data for the observed period (1990–1996) shows that positive surges (with high amplitude) occur both on the rising and falling tide with peaks before and after the tidal high water. Negative surge peaks also tend to avoid high water, especially on the rising tide. Figure 6.2.25 shows very interesting results regarding the distribution of negative surges relative to the tidal low water (LW). The peak of the negative surges (with high amplitudes as well) occurs 2–3 hours after the LW, which can be hazardous from navigational safety point of view. Finally, this nonlinear interaction between the tides and surges may significantly change the design return period for coastal defences against flooding.

### *6.2.7 Spatio-temporal analysis of the relationships between the meteorological forcing and the surges*

The regular tidal movements of the sea is continuously modified to a greater or lesser extent by the meteorological forcing due to the exchange of energy between the atmosphere and the sea at all time and space scales. As already discussed previously, these non-tidal residuals are usually called meteorological residuals or surges. Due to the chaotic dynamics of both the water levels and the surges (as already demonstrated), no

two surge events are exactly the same because small variations in weather patterns may produce quite different responses in the body of the sea water, particularly where there is a tendency for local standing waves and water-mass resonances and oscillations. Physically the atmosphere acts on the sea in two distinctly different ways. Changes in atmospheric pressure produce changes in the forces acting vertically on the sea surface which are felt immediately at all depths. Also, forces due to the wind stress are generated at and parallel to the sea surface: the extent to which they are felt at depths below the surface is determined by the time duration for which they act and by the density stratification of the water column, which control the downward transfer of momentum. Usually, due to their interaction, the effects of wind and air pressure on the surges are difficult to be identified and explained separately.

Several possible physical responses of the sea may be modelled by analytical simplified solutions to the hydrodynamic equations, but a global description of the complex relationships can be best studied by numerical modelling techniques. In order to accurately describe and predict the local surge events due to the meteorological forcing using numerical models, we need to have a very accurate description of the atmospheric changes (as initial and boundary conditions) on smaller time and spatial scales, which at this stage of the technological development are still missing. Furthermore, bearing in mind that both the meteorological dynamics and the sea level dynamics bear the hallmark of deterministic chaos, long-term predictability is not guarantied. Although it is usual to forecast the surges only in terms of the extreme high amplitudes, extreme negative surges may also be generated by the meteorological forcing and these have significant impact for the safe navigation of large vessels in shallow water, such as the approach channel at Hoek van Holland (recall Figure 6.2.25). Furthermore, both positive and negative extreme surges may be generated by the same meteorological forcing at different stages of its progression. The analysis of the historical extreme surges in the North Sea shows that large positive surges are often preceded by negative surges a day or so before, due to the pressure gradients (drops) travelling from the deep Atlantic to the shallow shelf waters, such as the damaging storm surge in 1953.

The global mechanism of surge generation due to the meteorological forcing in the North Sea along the Dutch coast is well-known (see for example, Heaps, 1983). This part of the North Sea is open to the North Atlantic ocean in the north so that the extratropical storms which travel across this entrance from west to east (see Figure 6.2.1) are able to set the water in motion with very little resistance from bottom friction. When these water movements are propagated into the North Sea they are affected by the earth's rotation and by the shallower water as they approach the narrowing region to the south towards the Dutch coast. These geostrophic disturbances, which travel from north to south like tides as Kelvin waves are sometimes called *external surges* in order to distinguish them from the movements and changes of sea level by the meteorological forcing, which are called *internal surges*.

The standard deviation computed on the time series of the meteorological residuals (surges) at the different locations along the Dutch coast (see Table 6.2.1) varies between $\sigma$=0.24 (m) at Euro platform to $\sigma$=0.38 (m) at Delfzijl, which shows very high values compared to the meteorological residuals at other parts of the world. The possible explanation of such a high variation in the meteorological residuals is the extensive area of very shallow water. The power spectrum of the residuals in Figure 6.2.18 shows that

although the astronomical tidal variations have been removed, there are still peaks of energy at the tidal frequencies due to the higher-order subharmonics in the shallow-water area and the interaction between the tides and surges. It is interesting to note that the diurnal band has no significant residual energy. This indicates that for the analysis of the meteorological residuals it is convenient to eliminate variations at frequencies above the diurnal tidal band. Figure 6.2.26 shows the relationship between the surge variations computed using a 72-hour filter and the inverse of the air pressure difference variations (related to the normal atmospheric pressure of 1013 mb). The low-pass filter was used in order to take into account the longer time-scales for the surge.

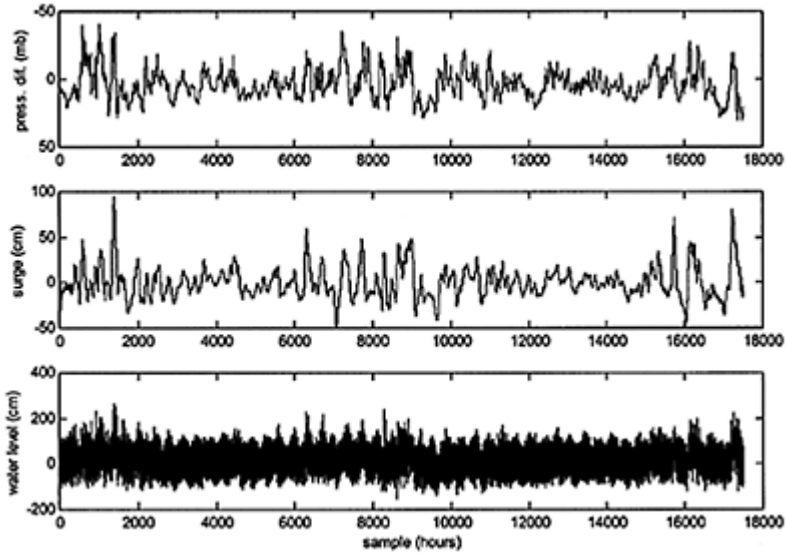



Figure 6.2.26. Variation of the meteorological residuals computed using low-pass 72-hour filter compared with the difference in air pressure at Hoek van Holland. The air pressure differences are plotted reversely.

This response of the sea level to the atmospheric pressure is the well-known inverted barometer effect. Figure 6.2.26 shows that the surge level changes are coherent with the inverse of the air pressure variations. For illustration: during a typical year atmospheric pressure may vary between values of 980 (mb) and 1030 (mb). Compared to the standard atmospheric pressure of 1013 (mb), this implies (by using simple analytical relationship $\Delta P_a = \rho g \Delta \xi$) a range of surge variations relative to the static sea level between +33 (cm) and −17 (cm). Figure 6.2.27 shows the cross-correlation and average mutual information between the air pressure and the surge at Hoek van Holland.

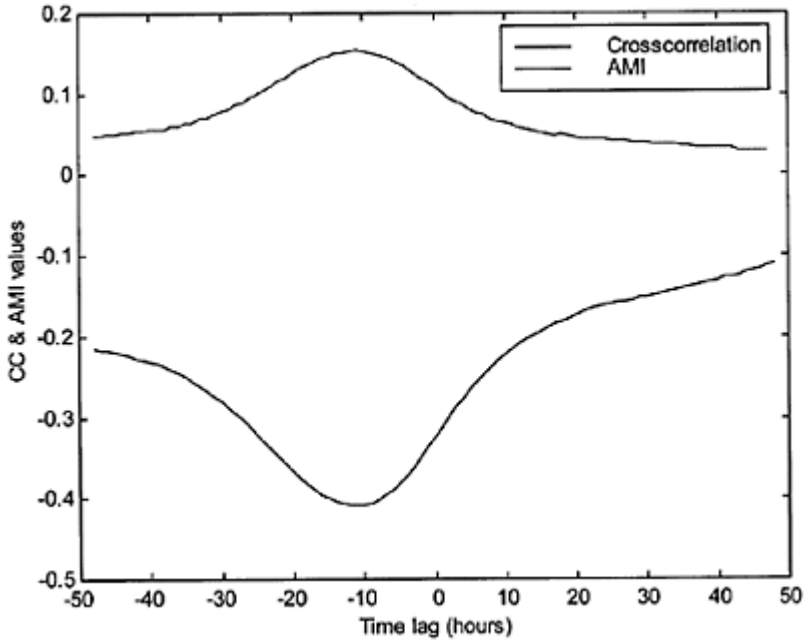



Figure 6.2.27. Cross-correlation (lower curve) and AMI (upper curve) between Air Pressure and Surge at Hoek van Holland tidal station.

Both relationships (cross-correlation and AMI) between the air pressure and the surge indicates that there is a substantial influence of the air pressure 10–12 hour ago on the future surge, which is reflected throughout the moving depressions towards East and North-East. The cross-correlation function is negative indicating the inverted barometric effect on the surge.

The exact inverted barometer response of the sea is seldom found in practice. As Figure 6.2.27 demonstrates the correlation coefficient is $r=-0.42$ which describes only part of the sea level variation. One reason for this is that the dynamic response of the shallow water to the movement of the atmospheric pressure fields and the wind effects is related to the movements of the air pressure fields. These variations, which are not accounted for by an inverted barometer response are most likely caused by the local winds and their interaction with the propagating pressure fields. The effect of the local winds on the currents at Hoek Van Holland within the framework of this research was studied in Hasan (2001) and further extended in this thesis with an analysis of the surges. The most effective wind direction for producing large surges is from south and southwest, which, in accordance to the Ekman transport to the right of the wind and the conservation of mass implies a corresponding build-up of the coastal sea level along with the long-shore current.

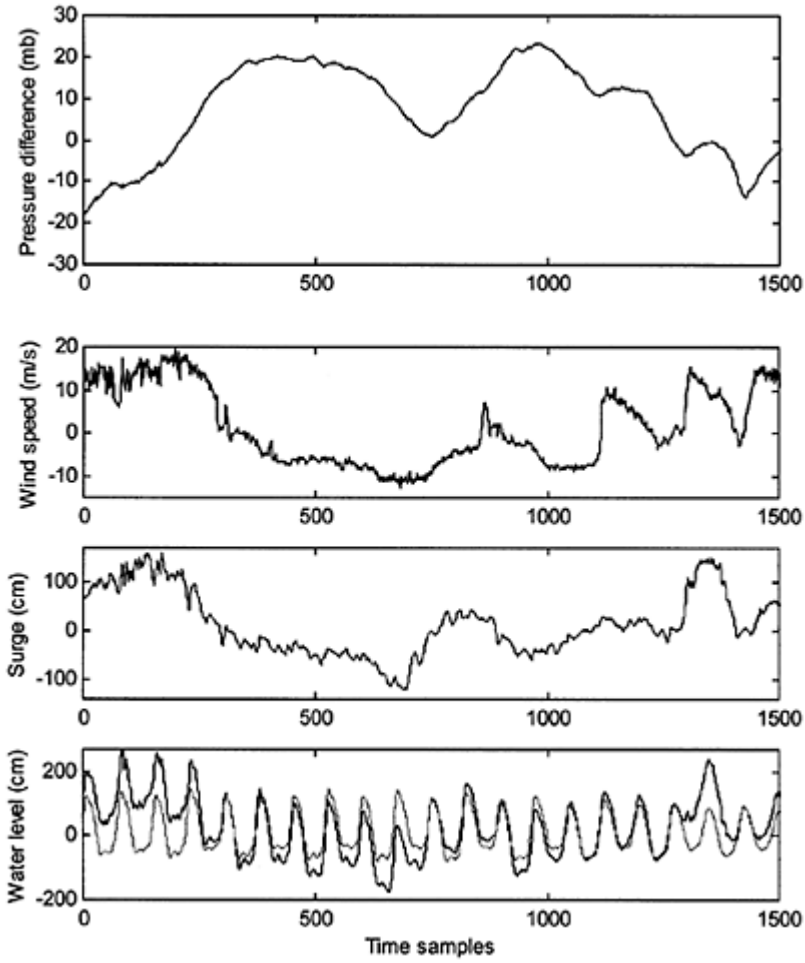



Figure 6.2.28. Relationship between the long-shore winds, surges, water levels, tides and air pressure differences at Hoek van Holland. The long-shore winds are projected on the southwest direction. Positive wind speed indicates that the wind component blows from southwest (210 degrees).

These wind-induced variations are called *locally generated surges* in order to distinguish them from the surges propagating freely as progressive waves. The nonlinear analysis

based on AMI showed that there is a good relationship between the long-shore and cross-shore southtwesterly winds and the surges. Figure 6.2.28 visualises the relationships between the long-shore components of the wind speed, surges, total water levels, astronomical tides and air pressure differences. The event presented in Figure 6.2.28 is from 1–January–1995 starting at 00:00 (144 samples represents 1 day). It is evident that recorded surges are quite extreme with the amplitude of about 2.7 (m), ranging between positive surges of 1.55 (m) and negative surges of −1.15 (m). The extreme positive surges (first 280 samples, i.e. duration of about 2 days) coincide with the negative pressure fields and strong long-shore winds (from the southwest) with a speed between 15–20 (m/s). The extreme negative surges (the next 2 days) are caused by the development of long-shore winds (towards the southwest) with a build-up of high pressure fields after the depression. In this case, according to the Eckman transport the coastal water is driven out from the coast, which with the combination of the high pressure fields induces extreme negative surges. Figure 6.2.29 shows the cross-correlation function between selected wind components and the surge at Hoek van Holland tidal station.

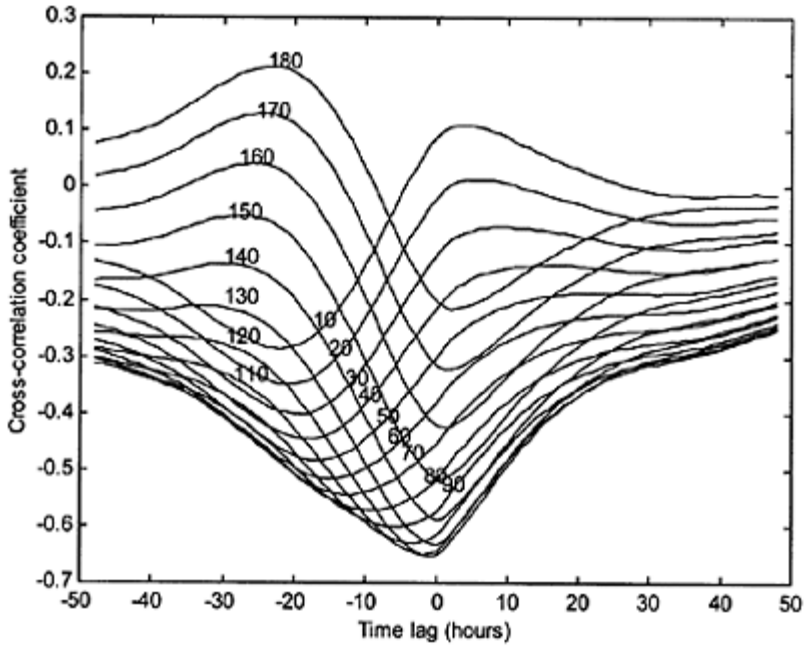



Figure 6.2.29. Cross-correlation function between wind speed component and the surge at Hoek van Holland tidal station. The numbers in the graph represent the angles of the wind component from the North. Positive correlation sighs indicate

winds from North-East and negative signs indicate wind from South-West.

The results presented in Figure 6.2.29 indicate that the strongest influence ($r$=−0.65) of the wind on the surge is generated by the cross-shore component of the wind (120 degrees from North, that is a North-Westerly wind). The time delay between this component of the wind and the surge is 1–1.5 hours. The along-shore component of the wind corresponding to the cross-shore component (30 or 210 degrees from North), has more a long-term impact on the surge generation due to the Eckman transport. A mass of water is moved towards the Dutch coast by the North-South winds which implies the generation of positive surges.

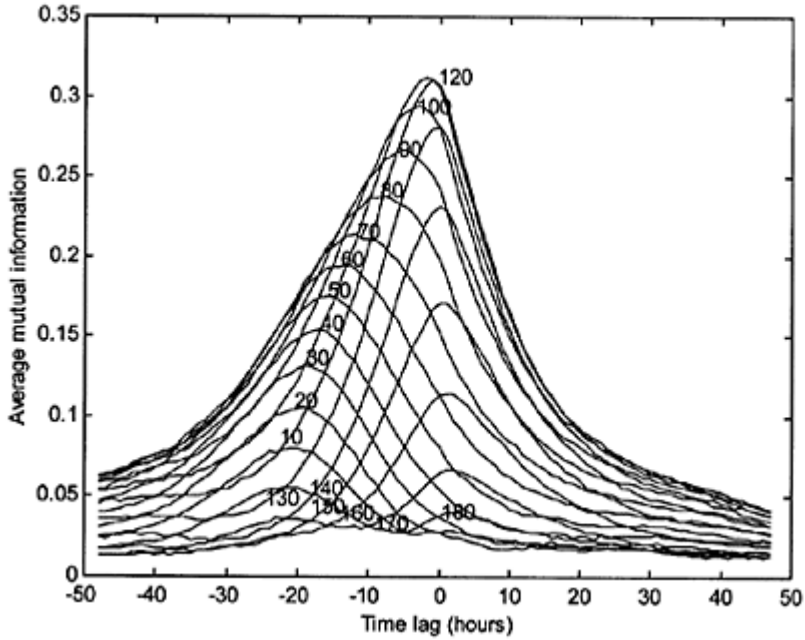



Figure 6.2.30. Average mutual information (bits) between wind speed components and the surge at Hoek van Holland tidal station. The numbers in the graph represent the angles of the wind component from the North.

Although the correlation coefficient for the along-shore winds is not significant (r=−0.41) these winds will certainly have a longer impact on the surges at Hoek of Holland (time delays between 10–20 hours) as presented and discussed in Figure 6.2.28. The average mutual information between the wind components and the surge at Hoek van

Holland show the same behaviour indicating a strong relationship between the cross-shore component of the wind (120 degrees from North, which is North-Westerly wind) and the surge for the lag of 1 hour; Figure 6.2.30.

The spatio-temporal analysis between the surge at Hoek van Holland and the wind components at the neighbouring stations K13 and EPF demonstrated similar findings; see Figure 6.2.31.

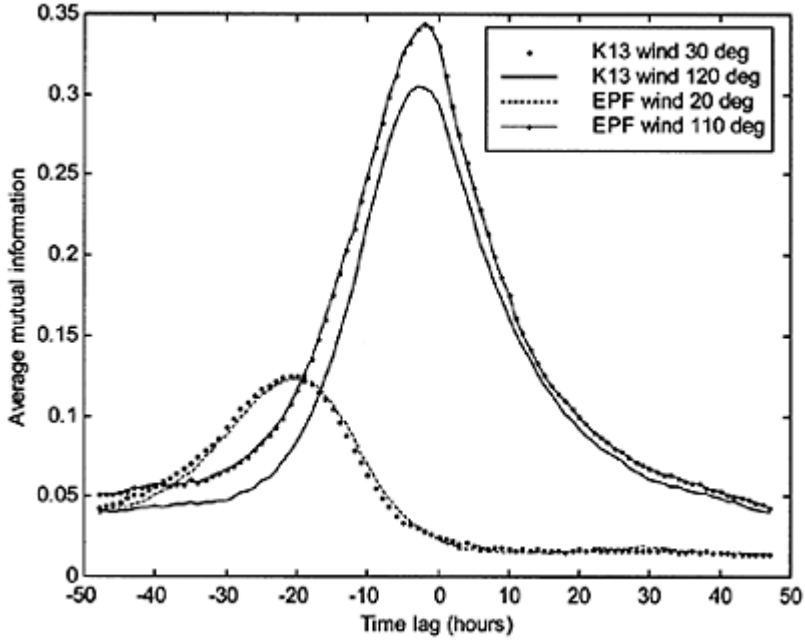



Figure 6.2.31. Average mutual information (bits) between wind speed components at K13 and Euro platforms and the surge at Hoek van Holland.

The AMI function presented in Figure 6.2.31 indicates that the strong influence of the wind at both K13 and Euro platform on the surge is due to the cross-shore component of the wind (120 and 110 degrees from North respectively). The time delays between the cross-shore component of the wind at K13 and Euro platform and the surge at Hoek van Holland are 2.5 and 2 hours respectively. Along-shore components of the wind (30 and 20 degrees from North) have a more long-term impact on the surge at Hoek van Holland (time delays between 18–20 hours).

Finally, the spatio-temporal relations between the surges at Hoek van Holland and the neighbouring stations (EPF, K13, IJmuiden and Vlissingen) were investigated. Both functions (AMI and cross-correlation) indicate very strong relationships between the surges at Euro platform and Vlissingen and the surge at Hoek van Holland. These surges precede the surge at Hoek van Holland by about 1 hour, thus carrying important

information for predictive modelling. In contrast, the surges at K13 platform and IJmuiden show the strongest relationships with the surge at Hoek van Holland 1–1.5 hours later. It is interesting to note that the linear cross-correlation function shows temporal dependencies between the surges for about 2 tidal cycles, whereas the average mutual information indicates a temporal dependence of about 1 tidal cycle.

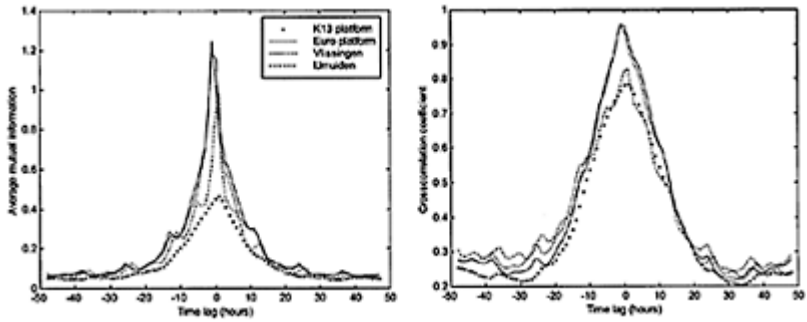



Figure 6.2.32. Average mutual information (left figure) and crosscorrelation (right figure) between the surges at the surrounding stations (K13, Euro platform, Vlissingen and Ijmuiden) and the surge at Hoek van Holland.

### *6.2.8 Local modelling and forecasting water levels and surges*

Based on the identified and reconstructed dynamics of both water levels and surges along the Dutch coast, an attempt was made to build an accurate short-term forecasting model utilising chaos theory and the notion of “dynamic neighbours” already elaborated above. Univariate local models using only information from the surge times series were constructed initially. This analysis was extended with multivariate local models in the reconstructed phase-space incorporating additional information for local mapping of the real dynamic neighbours, such as the tidal phases, the phase-locking phenomenon and the tide-surge interaction. Finally, the hybrid modelling framework—mixture of local models—elaborated in Chapter 5, has demonstrated the best forecasting performances. Herewith we summarise the prediction results (using different prediction horizons) for the surges at Hoek van Holland.

#### SURGE PREDICTION USING UNIVARIATE LOCAL MODELS

Adaptive local models (linear and polynomial) were used in the reconstructed phase space of the surge at Hoek van Holland to map the dynamics of the attractor. In this experiment only information from the surge time series was used to build the local

models. The sensitivity of the choice of the local approximation, the embedding dimension *(m)*, the time delay *(τ)* and the number of neghbours *(k)* were also investigated. Table 6.2.7 shows the available data set (training and testing) used to build and evaluate the local models. Tables 6.2.8, 6.2.9 and 6.2.10 summarise the performance of the univariate local models for the surge predictions.

Table 6.2.7. Training and testing parts of the surge data set (10min) for Hoek van Holand tidal station.

| | |
|---|---|
| Training set: | 1–Jan–1990->31–Dec–1994 (262943 samples) |
| Testing (unseen) sets: | 1–Jan–1995->31–Aug–1995 (34993 samples)—overall test period |
| | 1–Jan–1995->31–Mar–1995 (12960 samples)—stormy period, the most difficult for prediction |
| | 1–Jun–1995->31–Aug–1995 (13242 samples)—nonstormy period |

Table 6.2.8. Univariate model. Performance for the surge prediction based on univariate local 3rd order polynomial models using *10min* time series data (*m*=6, τ=20, *k*=35–50 for non-stormy period and *k*=10–15 for stormy period).

| | RMS Error (cm) for different prediction horizons (1 sample=10min) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 10min | 30min | 1 hour | 2 hours | 3 hours | 6 hours | 10 hours |
| **Overall test period (l–Jan–1995->31–Aug–1995)** | **2.89** | **4.55** | **7.27** | **11.94** | **14.57** | **19.66** | **25.23** |
| Stormy period (1–Jan–1995->31– | 3.10 | 5.15 | 9.52 | 15.35 | 17.55 | 23.32 | 27.78 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Mar–1995) | | | | | | | |
| Non-stormy period (1–Jun–1995->31–Aug–1995) | 1.80 | 2.95 | 4.01 | 5.98 | 7.24 | 8.99 | 10.63 |

In order to compare the surge predictions using 10min and hourly data further experiments were carried out using local models based on the hourly surge time series; see Table 6.2.9.

Table 6.2.9. Univariate model. Performance for the surge prediction based on univariate local 3rd order polynomial models using *1 hour* time series data ($m$=6, $\tau$=*6*, $k$=20–35 for non-stormy period and $k$=9–12 for stormy period).

| | RMS Error (cm) for different prediction horizons (1 sample=1 hour) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 10min | 30min | 1 hour | 2 hours | 3 hours | 6 hours | 10 hours |
| **Overall test period (l–Jan–1995->31–Aug–1995)** | / | / | **7.32** | **12.55** | **16.90** | **19.54** | **24.51** |
| Stormy period (1–Jan–1995->31–Mar–1995) | / | / | 9.74 | 15.51 | 18.92 | 23.12 | 27.18 |
| Non-stormy period (1–Jun–1995- | / | / | 4.55 | 6.07 | 7.69 | 8.81 | 10.15 |

| |
|---|
| >31–Aug–1995) |

The surge predictions were further compared with a univariate neural network model. The same reconstructed phase-space (input data) was used to train different NNs, using different architectures (MLP, modular, recurrent) and structures (number of hidden layer/nodes and transfer functions). Table 6.2.10 sumarise the results of the best performing NN, with a modular structure in this case.

Table 6.2.10. Univariate model. Performance for the surge prediction based on univariate MPL (6×4×1) neural network using *1 hour* time series data. Same input vectors from the reconstructed phase space were used.

| | RMS Error (cm) for different prediction horizons (1 sample=1 hour) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 10min | 30min | 1 hour | 2 hours | 3 hours | 6 hours | 10 hours |
| **Overall test period (l–Jan–1995->31–Aug–1995)** | / | / | **7.52** | **12.47** | **14.72** | **16.85** | **21.59** |
| Stormy period (1–Jan–1995->31–Mar–1995) | / | / | 9.81 | 16.04 | 17.96 | 22.49 | 29.45 |
| Non-stormy period (1–Jun–1995->31–Aug–1995) | / | / | 4.78 | 6.26 | 7.66 | 11.54 | 14.34 |

Figures 6.2.33, 6.2.34 and 6.2.35 further visualise the measured and predicted surge levels together with the errors for different prediction horizons based on the univarite local models in the reconstructed phase-space.

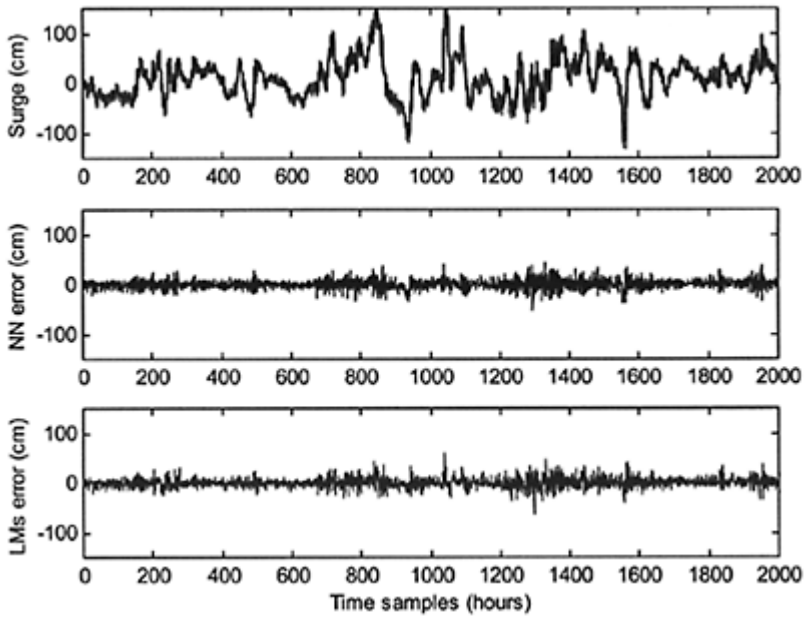



Figure 6.2.33. Prediction of the surges at Hoek van Holland based on hourly time series (solid grey line). The graph is zoomed at the stormy period (1–Jan–1995->31–Mar–1995). The prediction horizon is 1 hour. The overall RMSE for NN (blue dashed line) is 7.52cm and for LMs (red solid line) is 7.32cm. The bottom figures show the errors.

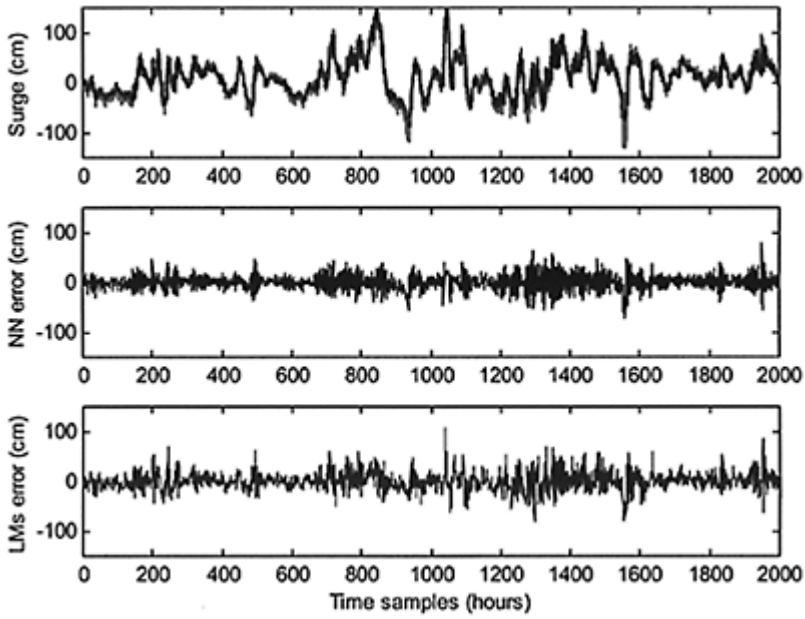



Figure 6.2.34. Univariate model. Prediction of the surges at Hoek van Holland based on hourly time series (solid grey line). The graph is zoomed at the stormy period (1–Jan–1995->31–Mar–1995) The prediction horizon is 3 hours. The overall RMSE for NN (blue dashed line) is 14.72cm and for LMs (red solid line) is 16.90cm. The bottom figures show the errors.

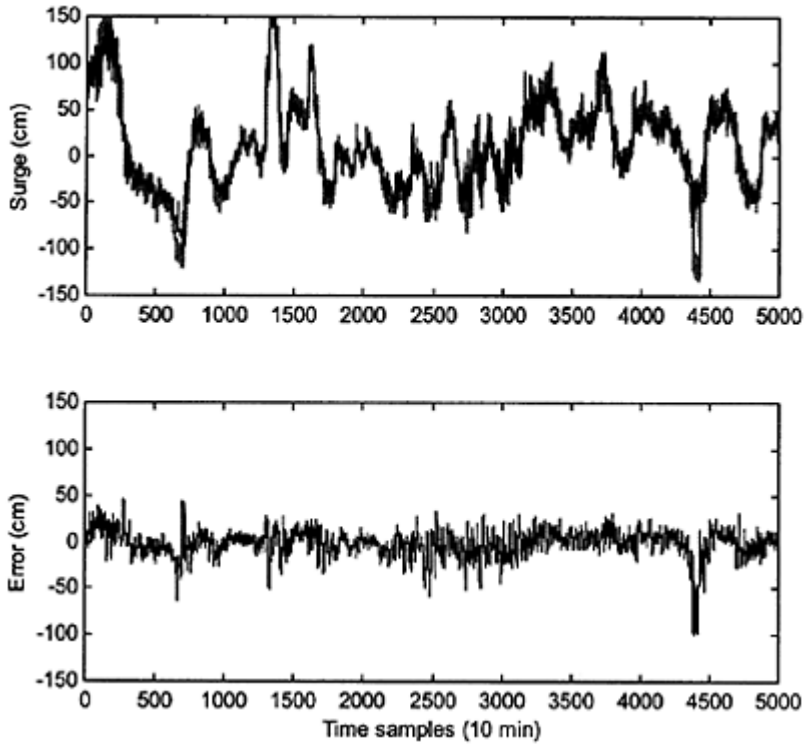



Figure 6.2.35. Univariate model. Prediction of the surges at Hoek van Holland based on 10min time series (solid grey line) using univariate local models. The graph is zoomed at the most variable pat of the time series (period 1–Jan–1995->4–Feb–1995). The prediction horizon is 3 hours with an overall RMSE of 14.57 cm. The bottom figure shows the prediction errors mostly due to a presence of phase error.

Figures 6.2.34 and 6.2.35 show that for longer prediction horizons using univariate models (surge time series only) there is a clear presence of a phase error. The amplitudes of the extreme positive surges are correctly predicted. However, the error on the extreme negative surges is larger with more pronounced phase error. In order to improve the surge predictions multivariate local models, including the meteorological forcing and different spatio-temporal information, were further explored.

## SURGE PREDICTION USING MULTIVARIATE LOCAL MODELS

Multivariate local models incorporating information on surge water levels (at Hoek van Holland and EPF), air pressure (difference), long-shore and cross-shore wind components, tidal phase class, phase-locking index, and tide-surge index, were tested with the main objective to improve the surge predictive accuracy for longer prediction horizons (3,6,10 hours). Root mean squared error (RMSE) was used as a model performance measure. Due to the computationally demanding task hourly data were used to construct these models. The multivariate phase-space reconstruction of the surge dynamics using hourly time series data was solved technically using the proposed methodology described in Section 3.3.8. The optimal reconstructed multivariate phase-space can be noted as:

$$\mathbf{Y} = \left\{ s_t^{hvh}, s_{t-6}^{hvh}, \ldots, s_{t-30}^{hvh}, w_t^{120^\circ}, \ldots, w_{t-3}^{120^\circ}, dp_t, dp_{t-6}, dp_{t-12}, s_t^{epf}, s_{t-1}^{epf}, \ldots, s_{t-5}^{epf} \right\}. \quad (6.2)$$

Table 6.2.11 presents the hourly data sets used in these experiments, and Table 6.2.12 and Table 6.2.13 summarise the model performances for both: the multivariate local models and the multivariate global ANN.

Table 6.2.11. Training and testing parts of the data set (hourly data) for Hoek van Holand tidal station.

| | |
|---|---|
| Training set: | 1–Jan–1990->31–Dec–1994 (43000 samples) |
| Testing (unseen) sets: | 1–Jan–1995->31–Aug–1995 (5832 samples)—overall test period |
| | 1–Jan–1995->31–Mar–1995 (2160 samples)—stormy period, the most difficult for prediction |
| | 1–Jun–1995->31–Aug–1995 (2208 samples)—nonstormy period |

Table 6.2.12. Multivariate model. Performance for the surge prediction based on *multivariate* local linear models using *1 hour* time series data ($m$=variable, $\tau$=variable, $k$=100–300 for non-stormy period and $k$=50–100 for stormy period).

| | RMS Error (cm) for different prediction horizons (1 sample=1 hour) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 10min | 30min | 1 hour | 2 hours | 3 hours | 6 hours | 10 hours |
| **Overall test period (1–Jan–1995->31–Aug–1995)** | / | / | **4.86** | **8.90** | **10.15** | **13.85** | **18.27** |
| Stormy period (1–Jan–1995->31–Mar–1995) | / | / | 5.55 | 10.66 | 14.25 | 18.96 | 24.19 |
| Non-stormy period (1–Jun–1995->31–Aug–1995) | / | / | 4.34 | 5.87 | 8.09 | 9.22 | 10.33 |

Table 6.2.13. Multivariate model. Performance for the surge prediction based on multivariate modular MPL (2 MPLs-19×16×8×1) neural network using *1 hour* time series data. Same input vectors from the reconstructed phase space were used.

| | RMS Error (cm) for different prediction horizons (1 sample=1 hour) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 10min | 30min | 1 hour | 2 hours | 3 hours | 6 hours | 10 hours |
| **Overall test period (1–Jan–1995->31–Aug–1995)** | / | / | **5.90** | **11.11** | **12.35** | **14.09** | **19.29** |
| Stormy period (1–Jan–1995->31–Mar–1995) | / | / | 7.38 | 13.72 | 14.12 | 19.31 | 25.18 |
| Non-stormy period (1–Jun–1995->31–Aug–1995) | / | / | 5.01 | 6.65 | 9.67 | 10.57 | 11.58 |

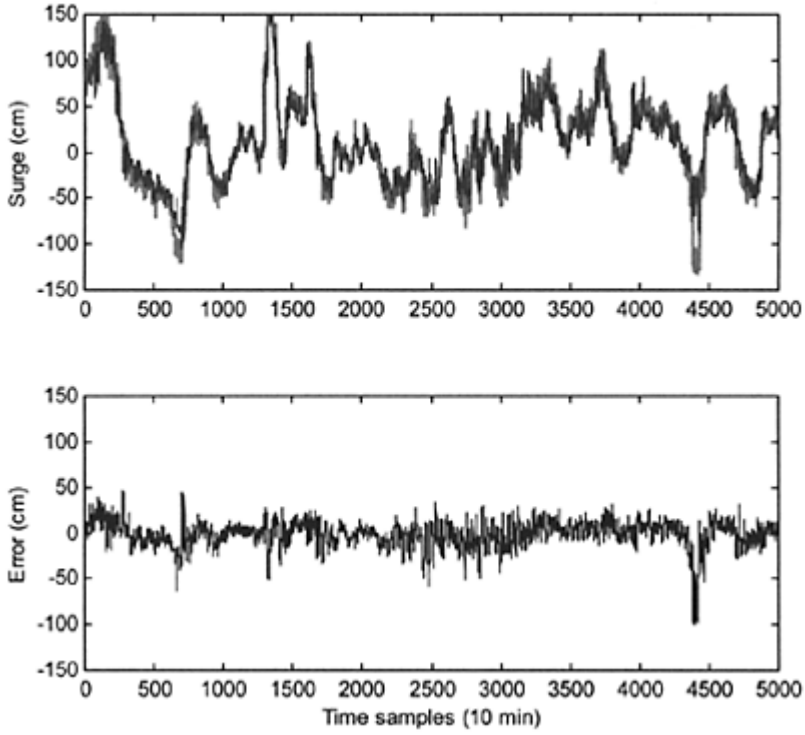



Figure 6.2.36. Multivariate model. Prediction of the surges at Hoek van Holland for the stormy period (1–Jan–1995->31–Mar–1995) based on hourly time series (solid grey line). The prediction horizon is 1 hour. The overall RMSE for multivariate ANN (blue dashed line) is 5.90 cm and for multivariate LMs (red solid line) is 4.86 cm. The bottom figures show the errors.

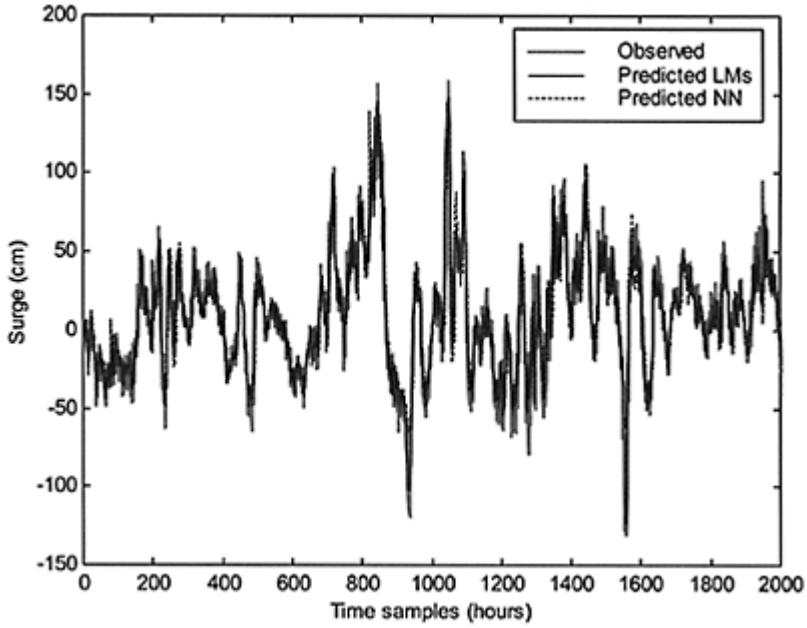



Figure 6.2.37. Multivariate model. Prediction of the surges at Hoek van Holland for the stormy period (1–Jan–1995->31–Mar–1995) based on hourly time series (solid grey line). The prediction horizon is 3 hours. The overall RMSE for multivariate NN (blue dashed line) is 12.35cm and for multivariate LMs (red solid line) is 10.15cm. The phase error is not present any more.

The surge predictions based on the multivariate local models, incorporating information on the meteorological forcing and additional knowledge expressed in a form of rules for selecting proper dynamical neighbours in the reconstructed phase-space, have shown significant improvements compared to the univariate local models. It is further evident that the existing phase error is not present any more. Finally, the multivariate local models showed better short-term (up to 10 hours ahead) predictive performances in comparison with the optimal neural network for the same data sets, especially in demonstrating capabilities for a more accurate prediction of the extreme negative surges; see Figure 6.2.37. The additional experiments, carried out for longer prediction horizons (up to 24 hours ahead), showed better performances of the global neural network model,

since it is capable of capturing the global input-output relationships based on the available time series data.

## SURGE PREDICTION USING MIXTURE OF MODELS (HMMMS)

The local modelling experiments showed that there are clear regions (dynamical regimes) on the attractor of the reconstructed phase-space that can be modelled using different local models with different parameters ($\tau$, $m$ and $k$), i.e. capacity. For illustration, Figure 6.2.38a shows the sensitivity of the choice of the number of neighbours $k$ for the local multivariate linear models on the surge predictions. Thus a mixture of local models using the data-driven modelling framework elaborated in Chapter 5 was constructed specialising on different surge dynamics, and showed an improved forecasting performance. Table 6.2.14 and Figures 6.2.38, 6.2.39, 6.2.40, 6.2.41 and 6.2.42 summarise the surge predictions using the mixture of models approach.

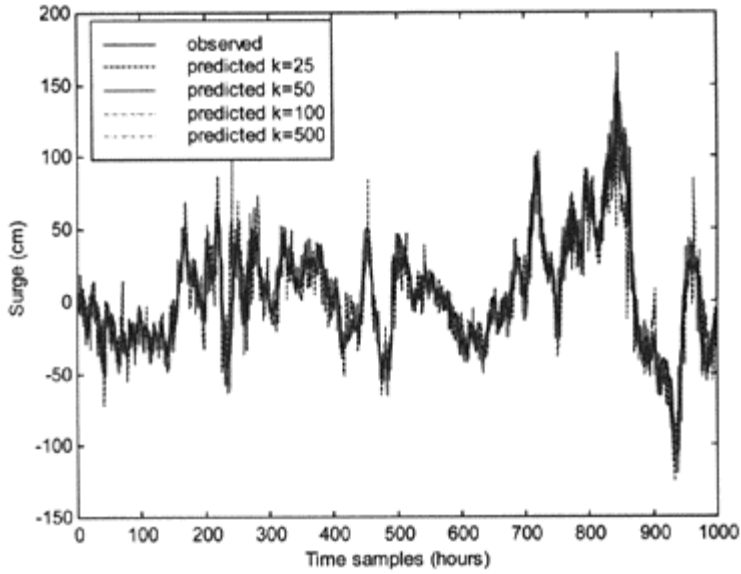



Figure 6.2.38a. Sensitivity of the choice of the number of "dynamic" neighbours on the surge prediction.

Table 6.2.14. Mixture of models. Performance for the surge prediction based on mixture of multivariate local linear models using *1 hour* time series data.

| | RMS Error (cm) for different prediction horizons (1 sample=1 hour) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 10min | 30min | 1 hour | 2 hours | 3 hours | 6 hours | 10 hours |
| **Overall test period (1–Jan–1995->31–Aug–1995)** | / | / | **4.35** | **8.01** | **9.57** | **12.65** | **17.35** |
| Stormy period (1–Jan–1995->31–Mar–1995) | / | / | 5.01 | 9.69 | 12.88 | 16.61 | 22.86 |
| Non-stormy period (1–Jun–1995->31–Aug–1995) | / | / | 3.98 | 5.57 | 7.98 | 9.05 | 10.20 |

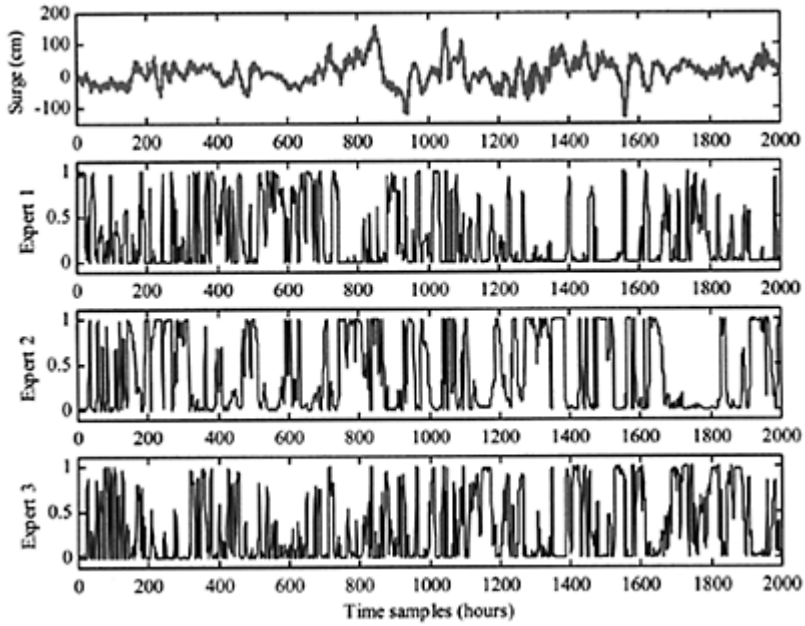



Figure 6.2.38. Activation function of each local model (expert) zoomed at the stormy period (1–Jan–1995->31–Mar–1995). Prediction horizon is 1 hour. Each expert is a multivariate local linear model using different embedding dimension *m* and different number of neighbours *k*. The final prediction is a soft combination of the prediction of the three experts in this case. For determining the optimal number of experts, a cross-validation data was used as a part of the training data set.

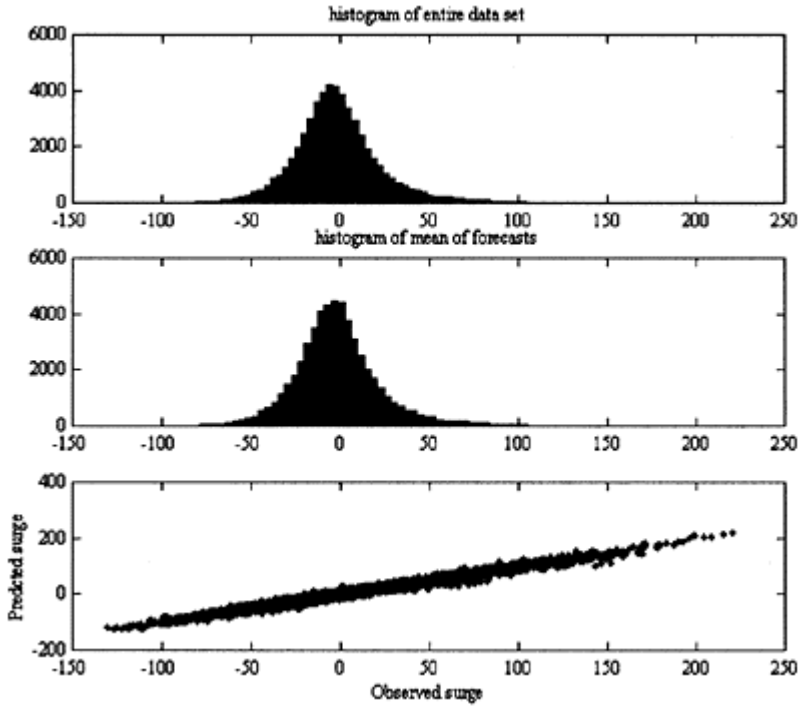



Figure 6.2.39. Performance of the mixture of models expressed through the observed and predicted density distributions for 1 hour prediction of the surge at Hoek van Holland.

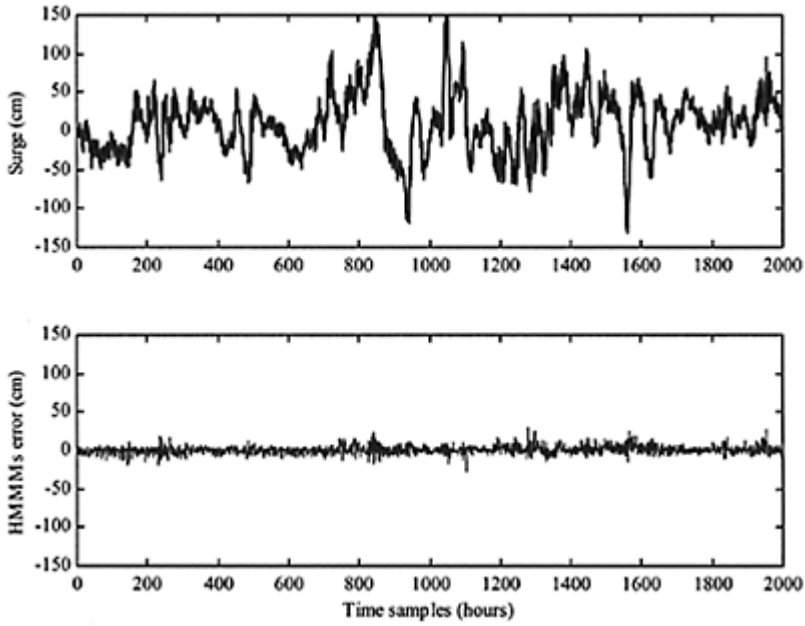



Figure 6.2.40. Mixture of models. Prediction of the surges at Hoek van Holland zoomed at the stormy period (1–Jan–1995->31–Mar–1995) based on hourly time series (solid grey line). The prediction horizon is 1 hour. Mixture of local multivariate models (blue solid line) were used. The overall RMSE is 4.35 cm. The bottom figure shows the errors.

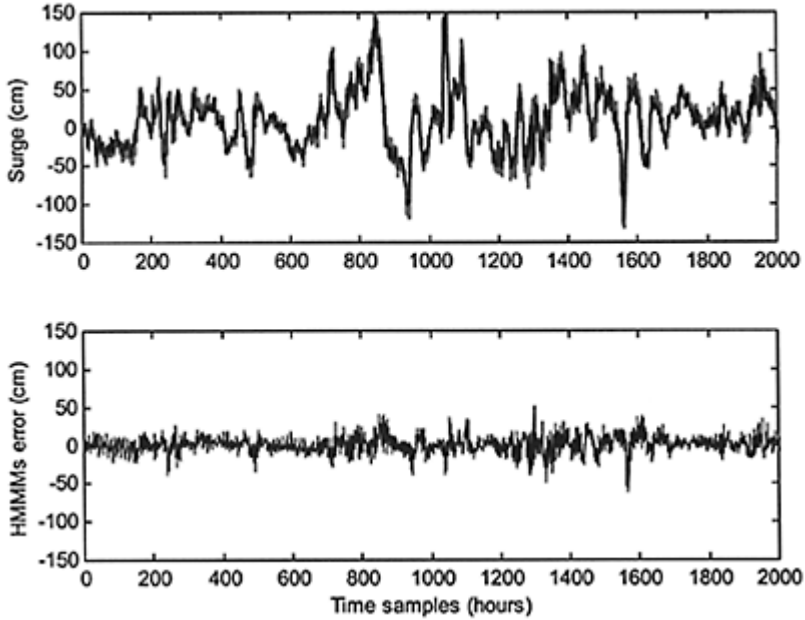



Figure 6.2.41. Mixture of models. Prediction of the surges at Hoek van Holland zoomed at the stormy period (1–Jan–1995->31–Mar–1995) based on hourly time series (solid grey line). The prediction horizon is 3 hours. Mixture of local multivariate models (blue solid line) were used. The overall RMSE is 9.57cm. The bottom figure shows the errors.

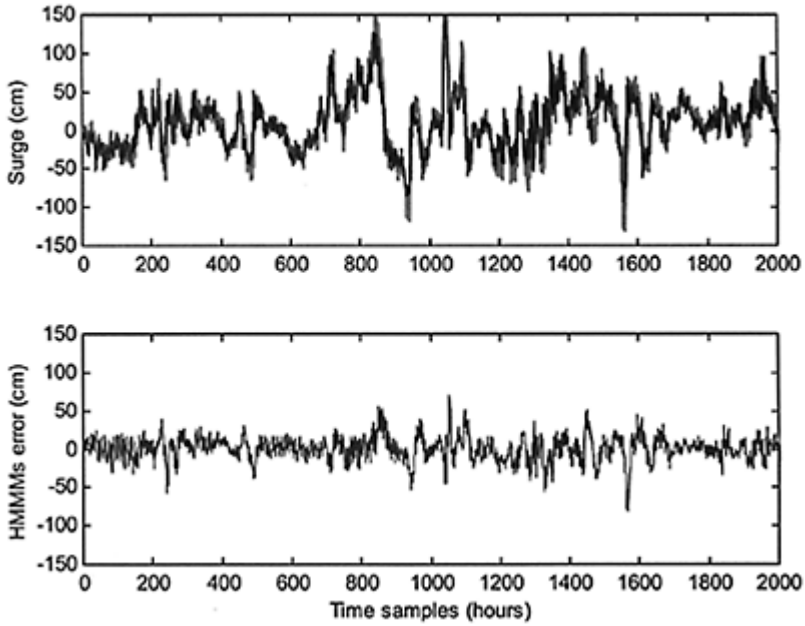



Figure 6.2.42. Mixture of models. Prediction of the surges at Hoek van Holland zoomed at the stormy period (1–Jan–1995->31–Mar–1995) based on hourly time series (solid grey line). The prediction horizon is 6 hours. Mixture of local multivariate models (blue solid line) were used. The overall RMSE is 12.65cm. The bottom figure shows the errors.

SURGE PREDICTION USING MIXTURE OF MODELS (HMMMS) AND INCLUDING FUTURE METEOROLOGICAL INFORMATION

The mixture of models framework showed improved surge forecasting performances using real prediction mode (that is, no future information was included in the models-experts). In order to test the performance of the mixture of models framework in real operational mode, future (predicted) meteorological information provided by the global climate model (HIRLAM) was included into the model. Due to the lack of a time series of predicted meteorological information (air pressure and wind fields along the Dutch coast) for the period of 1990–1996, and based on the overall errors of the HIRLAM model for prediction of the air pressure and wind fields for 1999 and 2000, the predicted data for 1995–1996 was assumed to be the measured data disturbed with 10–20% of the

variances of the both time series. Future surge information was included using predicted surges based on univariate local models. Table 6.2.15 and Figures 6.2.43, 6.2.44, and 6.2.45 summarise the surge predictions using the mixture of models approach including future meteorological information.

Table 6.2.15. Mixture of models including future meteorological information. Performance for the surge prediction based on mixture of multivariate local linear models using *1 hour* time series data.

| | RMS Error (cm) for different prediction horizons (1 sample=1 hour) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 10 min | 30 min | 1 hour | 2 hours | 3 hours | 6 hours | 10 hours |
| **Overall test period (1–Jan–1995->31–Aug–1995)** | / | / | **3.85** | **5.32** | **6.12** | **8.02** | **10.50** |
| Stormy period (1–Jan–1995->31–Mar–1995) | / | / | 4.01 | 5.66 | 6.89 | 8.75 | 11.62 |
| Non-stormy period (1–Jun–1995->31–Aug–1995) | / | / | 3.18 | 4.37 | 5.23 | 6.56 | 7.89 |

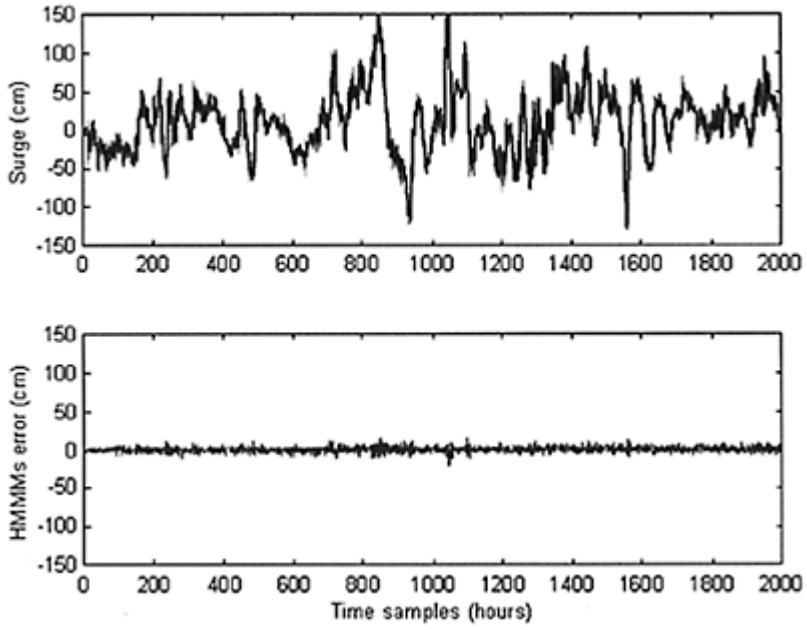



Figure 6.2.43. Mixture of models including future meteorological information. Prediction of the surges at Hoek van Holland zoomed at the stormy period (1–Jan–1995->31–Mar–1995) based on hourly time series (solid grey line). The prediction horizon is 1 hour. Mixture of local multivariate models (blue solid line) were used. The overall RMSE is 3.85cm. Bottom figure shows the error.

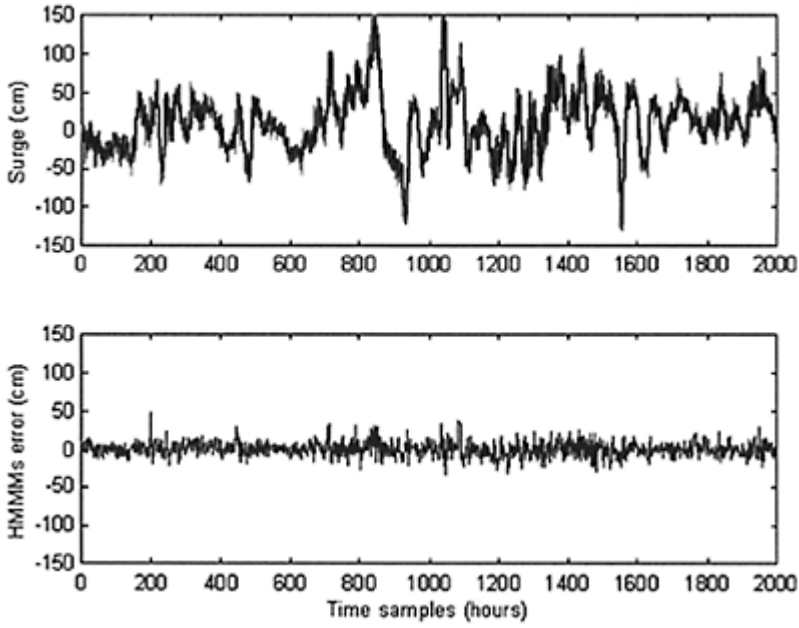



Figure 6.2.44. Mixture of models including future meteorological information. Prediction of the surges at Hoek van Holland zoomed at the stormy period (1–Jan–1995->31–Mar–1995) based on hourly time series (solid grey line). The prediction horizon is 6 hours. Mixture of local multivariate models (blue solid line) were used. The overall RMSE is 8.02cm. Bottom figure shows the error.

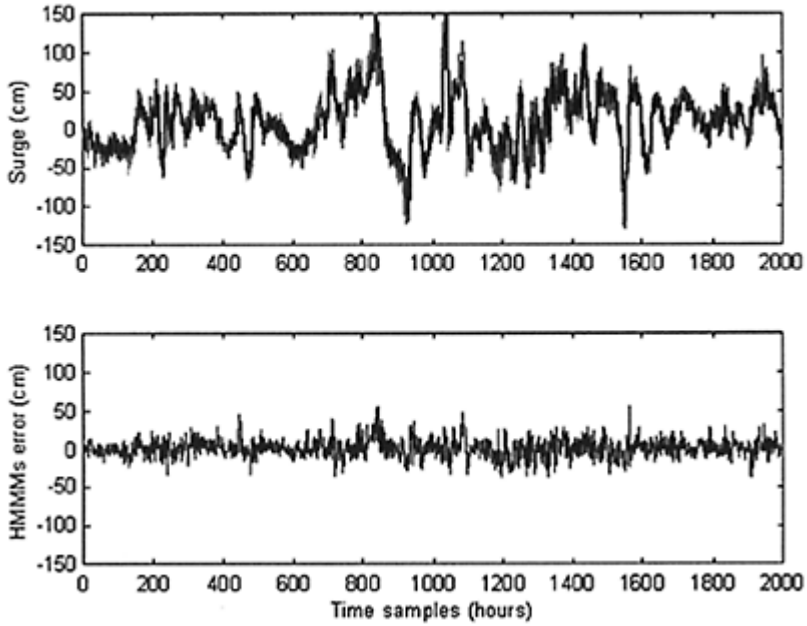



Figure 6.2.45. Mixture of models including future meteorological information. Prediction of the surges at Hoek van Holland zoomed at the stormy period (1–Jan–1995->31–Mar–1995) based on hourly time series (solid grey line). The prediction horizon is 10 hours. Mixture of local multivariate models (blue solid line) were used. The overall RMSE is 10.50cm. Bottom figure shows the error.

The performances of the mixture of models framework with future meteorological information were compared with the results from the numerical model WAQUA, which is currently in use for operational forecasting of the surges, currents and wave heights along the Dutch coast. Some average results from WAQUA model for 1995 and 1999 in hindcasting mode (model re-run with forecasted and assimilated meteorological information: air pressure and wind fields) are (source: RIKZ and DNZ):

- Overall surge prediction: RMSE=9–10cm for horizons of 6–12 hours
- Surge prediction for the stormy period: RMSE=12–15cm for horizons of 6–12 hours
- Surge prediction for the non-stormy period: RMSE=6–8cm for horizons of 6–12 hours.

The surge forecasting results at Hoek van Holland generated by the mixture of models framework using local multivariate models are comparable with the results from the WAQUA numerical model in hindcasting mode. It is worth mentioning that the CPU time for the mixture of models framework is the order of seconds in operational forecasting mode.

Finally, an experiment of surge prediction using the measured future meteorological information (air pressure and winds at HVH, k13 and EPF) was carried out that demonstrated a very low error on the surge predictions. This implies that the set-up of the mixture of models framework and the parameters for the local multivariate models together with the procedure of selecting the best neighbours in the reconstructed phase-space were properly incorporated. Furthermore, these results indicate that the uncertainties in surge prediction are largely dependent on the meteorological forcing.

### *6.2.9 Assessment of the local entropy and predictability of the surge dynamics*

The estimated geometrical and dynamical invariants of the reconstructed surge dynamics, (see Table 6.2.3), such as the correlation dimensions, Lyapunov exponents and Kolmogorov-Shinai entropies are average measures about the self-similarity, predictability and the complexity of the analysed dynamical system. What is more interesting from an operational forecasting perspective is quantifying in which cases the predictions of such complex dynamics in the future can be reliable and certain and in which cases they are uncertain. The basic methodology to examine this problem, based on the theory of nonlinear dynamics and conditional entropies as already elaborated in Section 3.3, was further applied to the surge dynamics at Hoek van Holland. Using the methods of symbolic dynamics and clustering techniques the reconstructed trajectory of a dynamical system in phase-space is mapped to a string (sequence) of letters on a certain alphabet. This string of letters is further analysed using information theoretical methods.

A direct application of the conditional entropy concepts requires a symbolic representation of the available real valued data $X_t$. This is achieved by introducing a finite partition $\prod$, which divides the reconstructed continuous phase-space $\mathbf{Y}$ (Eq.6.2) into $\lambda$ disjoint sets. Each set is labelled with a symbol (letter) *Ai* from the alphabet A. In such a manner, the resulting symbolic sequence represents a discrete (coarse-grained) description of the time evolution of the dynamical system. The finite partition $\prod$ of the reconstructed multivariate phase-space of the surge dynamics into $\lambda$ disjoint sets was done using the Bayesian nonsupervised clustering algorithm known as AutoClass (Stutz and Cheeseman, 1994). This is elaborated in Velickov and Solomatine (2000). The main challenge in creating the partition $\prod$ is to find a good balance between a finer partition based on the relationships that underline the surge dynamics and the statistical significance of the entropy analysis due to the finite length of the times series of observables. In other words, the statistics of the entropy analysis is biased by the length $\lambda$ of the alphabet (number of disjoint sets of the partition) since the required length of the time series to compute the entropies is of order $O(\lambda^n)$. To be specific, based on the cluster analysis of the reconstructed phase-space of the surge dynamics at Hoek van Holland seven clusters were found to give best results in terms of the log of the relative marginal probability of the clustering model given the data. Each cluster, which maps the

individual point in phase-space to the symbolic sequence, was labelled using the following alphabet (integer numbers in this case) A= {0,1,2,3,4,5,6}. Figure 6.2.46 shows the real-valued surge time series together with the corresponding sequence of symbols.

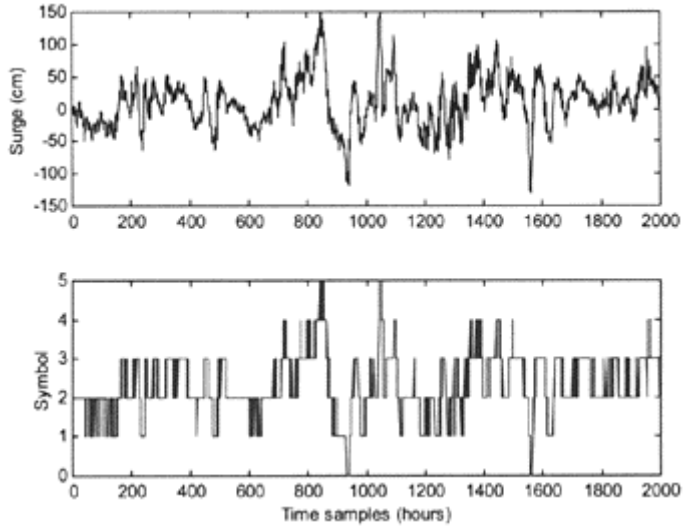



Figure 6.2.46. Surge time series and corresponding sequence of symbols zoomed at a stormy period (period 1–Jan–1995->31–Mar–1995).

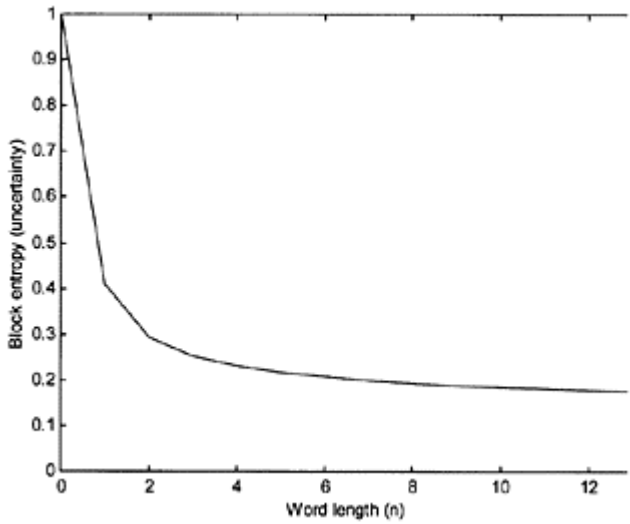



Figure 6.2.47. Average block (*n*-gram) entropy (uncertainly) as a function of the word length *n*. Beyond *n*=6 the calculation of the block entropy is not reliable due to insufficient statistics.

The generated sequence of symbols mapping the surge time series was further analysed using the entropy concepts described in Section 3.3.5 and elaborated in Velickov *et. al* (2003). The analysis of the average block (Eq.3.72) and conditional entropies (Eq.3.73) for different word length *n* is presented in Figure 6.2.47. We see that the average predictability of the surge dynamics over the complete data set is good and is higher than 80%. However, the average dynamic uncertainties do not give much insight into the variability of the local order and predictability in different dynamic conditions (regimes).The result of the calculation of the local uncertainty $h_n^{(1)}$ $(A_1 \ldots A_n)$ and predictability $r_n^{(1)}$ $(A_1 \ldots A_n)$ for the next hour following behind an observed section $A_1 \ldots A_n$ of the trajectory, according to Eq.(3.) and Eq.(3.), for $n$=2 is presented in Figure 6.2.48. We see that the predictability which is based on the regularities found in the sequence for a memory ($n$=2) varies between 0.63–1.0. Figure 6.2.49 shows both the predictability of the 3$^{rd}$ and the 5$^{th}$ symbol (3 time steps ahead), calculated according to Eq.(3.) and Eq.(3.), following an observed section of the trajectory in phase-space for $n$=2.

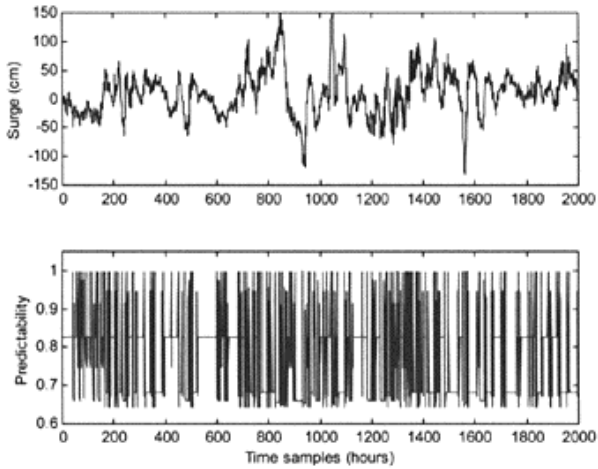



Figure 6.2.48. Surge time series and local predictability (uncertainty) $r_2$ of the prediction of the of the 3rd symbol based on the 2 preceding symbols in phase-space.

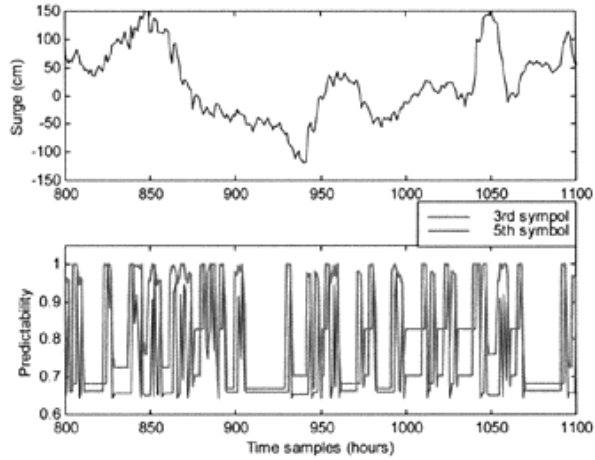



Figure 6.2.49. Local predictability (uncertainty) $r_2$ of the prediction of the 3rd and the 5th symbol based on the 2 preceding symbols in phase-space. The figure is zoomed at the storm surge in Jan 1995.

As presented in Figure 6.2.49, the local uncertainty of the 5$^{th}$ symbol (prediction horizon 3 hours) is higher than the 3$^{rd}$ symbol (prediction horizon 1) thus implying lower predictability. The optimal length of the section of the trajectory (memory of the system) based on which the local uncertainty and predictability are calculated can be assessed by looking at the mutual information (Eq. 3.) and autocorrelation function. Figure 6.2.50 indicates that the first minimum of the average mutual information is at lag $\tau$=6–8 hours whereas the exponentially decaying autocorrelation function reaches its value of 1/*e* (Tsonis and Elsener, 1988) (1/2.7182=0.37) at lag $\tau$=13–15 hours, indicating some memory of the dynamics on a tidal cycle level. Since the reconstructed phase-space of the surge dynamics includes information on a tidal cycle level (Eq.6.2), the analysis of local uncertainty and predictability based on sections of the trajectory with length $n$=4 is considered as sufficient and with the required statistics.

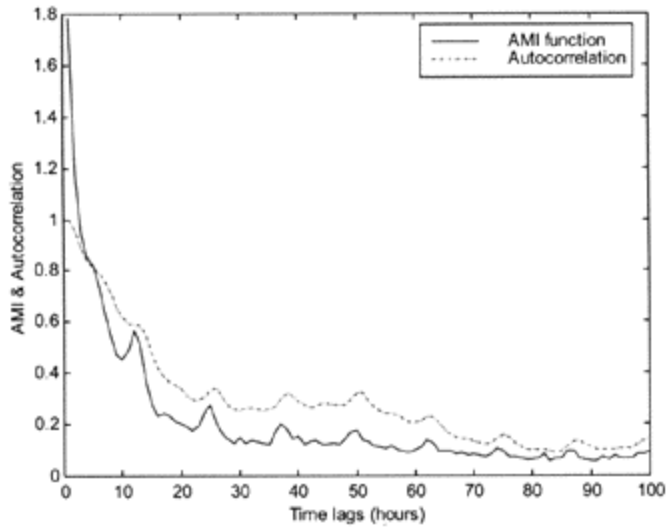



Figure 6.2.50. Average mutual information and autocorrelation functions for the hourly surge time series at Hoek van Holland.

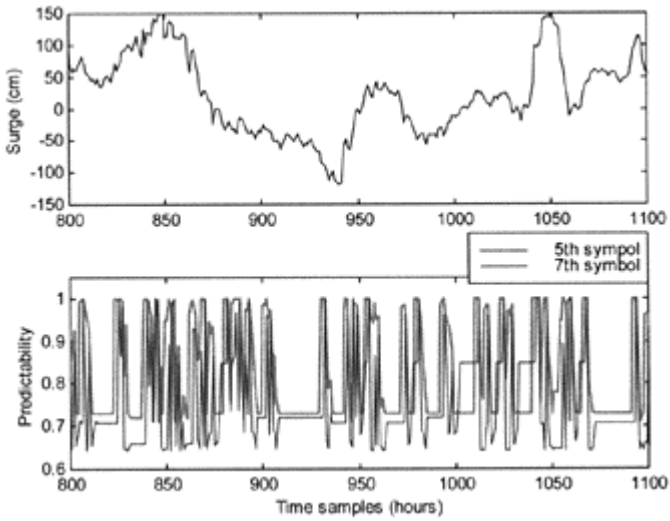



Figure 6.2.51. Local predictability (uncertainty) $r_4$ of the prediction of the $5^{th}$ and the $7^{th}$ symbol based on the 4 preceding symbols in phase-space.

The results of the calculation of the local uncertainty and predictability for the next hour following an observed section of the trajectory in phase-space for $n$=4 are presented in Figure 6.2.51. Behind certain patterns of surge dynamics the local predictability reaches 95–100%. This is evident for the patterns of type “3345” which indicate the dynamics of positive surges, patterns of type “1123” which indicate the transition from negative towards positive surges and patterns of type “3344” which indicate persistency in the positive surges. These types of surge patterns are driven by the meteorological forcing and the tidal motion. However, predictability of the patterns of type “4565”, which indicate positive surge peaks, is very uncertain and drops to between 75–80%. Similarly, predictability for the surge patterns of type “2101”, which indicate peaks in the negative surges, is below 72%. It is interesting to note that the lowest predictability shows patterns of type “2211”, “1111” and “2223”, which indicates persistency of small negative and positive surges. These surges are due to the shallow-water dynamic processes, which cause nonlinear interaction between the different tidal constituents, appearance of double low water and distorted high water, and a phase-locking phenomenon on different time scales between the different daily tidal constituents as already demonstrated and elaborated. Finally, the analysis also shows that the local predictability of the positive extreme surges in general is better than the negative extreme surges, which on the other hand are very important for the everyday safe ship navigation and guidance processes.

### *6.2.10 Conclusions*

In summary, the following major conclusions resulted from this case study:

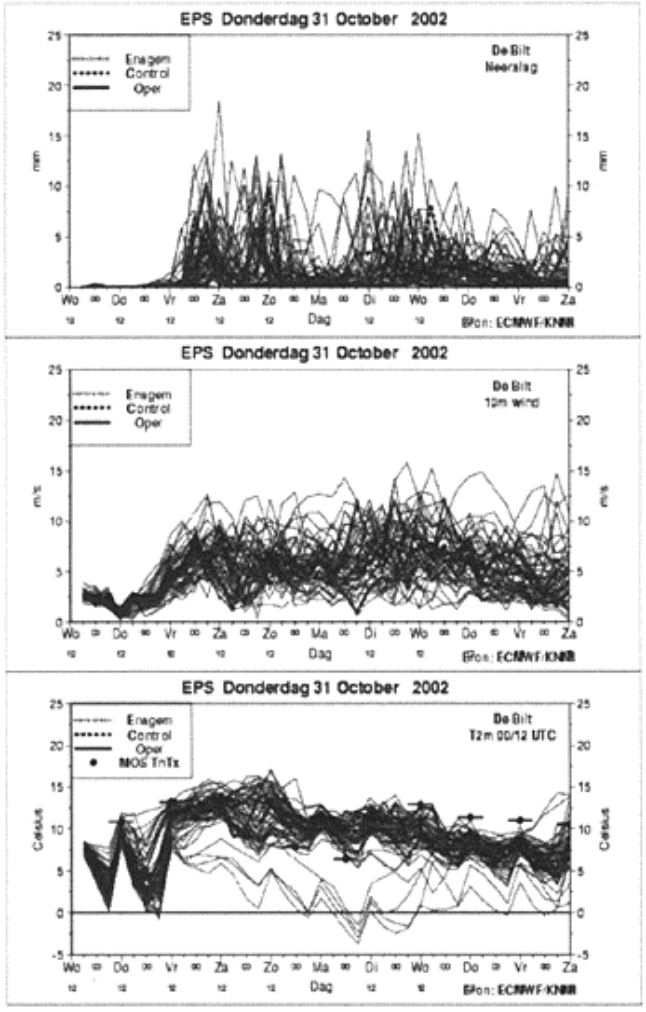
1. Based on the nonlinear analysis, phase-space reconstruction and estimation of various geometrical and dynamical invariants, the dynamics of both water levels and surges along the Dutch coast can be characterised as *deterministic chaos*. The presence of the chaotic dynamics together with the positive Lyapunov exponents implies that there are limits of predictability for any model (refer to Table 6.2.2 and Table 6.2.3). However, reliable short-term predictions are possible.
2. The chaotic behaviour occurs because water levels and surges, including astronomical contributions and the contributions from the meteorological forcing, are the result of a complex, coupled nonlinear dynamical system. The analysis of the shallow-water dynamics has demonstrated and explained the appearance of the double low water and the distortion of the duration of the high waters.
3. The Lyapunov exponents and the entropies have significant ramifications for numerical models that are based on solutions of the hydrodynamic equations of motion. The implication of the presence of deterministic chaos in surge dynamics is that estimates of future behaviour are very sensitive to mathematical formulations and assumptions, the choice of various coefficients and parametrisation, and the system's current state may be also inadequately modelled or measured. The main implication is that improvements in forecasting may require significant improvements in the accuracy of the terms, coefficients and the measurements, which are used as initial and boundary conditions, especially in the meteorological forcing. Data assimilation techniques, based on very accurate measured data may contribute to the improvement of the prediction performances.
4. Taking into account the presence of deterministic chaos in the water level and surge dynamics, a mixture of multivariate adaptive local modelling in the reconstructed phase-space of the dynamical system, which uses information from the real dynamical neighbours, has demonstrated good capability for reliable short-term predictions. For the Hoek van Holland location, the overall prediction error for the surge 10 hours ahead is about 10.5cm. For stormy sea dynamics the prediction error is about 12 (cm) and about 8 (cm) for non-stormy sea dynamics (the test data was taken from the period 1.01.95–31.08.95).
5. Identification and selection of proper dynamical neighbours from the historical time series data is the key issue in the local modelling approach adopted in this work. The dynamical selection of the types and the number of neighbours in the modelling procedure indicates that there are different dynamical regimes present in the sea dynamics that may be modelled using different types of models (e.g. local models, neural networks, etc.). Herewith the mixture of models framework showed the best predictive performances.
6. The local uncertainty analysis is an appropriate technique for studying the predictability of the surge dynamics. Although the overall predictability is about 80%, there exist certain dynamical situations when the predictability is much better than the average predictability and certain dynamical situations when the predictability is quite low, especially for the negative surges.

7. Chaos theory can serve as an efficient tool for accurate and reliable short-term predictions of water levels in order to support decision-makers in ship navigation.

## 6.3 Chaos in rainfall dynamics

### *6.3.1 Introduction*

The application of the theory of nonlinear dynamics associated with the concept of strange attractors for the description and modelling of deterministic chaos in hydrology has been gaining considerable interest in the last decade (Hense, 1987; Rodriguez-Iturbe et al., 1989; Sharifi et al., 1990; Tsonis et al., 1993; Jayawardena and Lai, 1994; Georgakakos et al., 1995; Koutsoyiannis and Pachakis, 1996; Sivakumar et al., 1998, 1999; Sivakumar 2000). Reconstruction of the dynamics of the hydrological system based on observables is seen as an important and integral part for understanding of the structure of particular hydrological process and gaining new knowledge in order to complement physically-based modelling. The chaotic nature of the weather has been demonstrated by numerical experiments with global circulation models. Studies using the most sophisticated global circulation models demonstrate that forecasts have a sensitive dependence on their initial conditions. Figure 6.3.1 shows some of the outputs of the global circulation model—HIRLAM used by the Royal National Meteorological Institute (KNMI). An Ensemble Prediction System (EPS) based on 50 model runs using slightly different initial conditions is used for the operational forecasting of the weather including precipitation.




Figire 6.3.1. An Ensemble Prediction System (EPS) based on global circulation model (station De Bilt, source KNMI). The upper graph shows the precipitation forecast for 10 days. Middle graph shows the wind speed at 10 m above the ground and the bottom graph shows the temperature at 2 m above the ground.

As a result, even if future global circulation model perfectly simulates the dynamics of the atmosphere, the predictability (as discussed in the previous section) of weather variables (especially precipitation) would approach zero for forecasts beyond 7–10 days (see for e.g. Schubert and Suarez, 1989). Detailed analyses of simplified atmospheric models have been used to study the underlying characteristics of chaotic behaviour (Lorenz, 1963, 1982, 1991; Ott, 1993).

Analyses based on numerical models and analysis of time series of observables by the theory of nonlinear dynamics and chaos mathematics, provide decisive evidences regarding the existence or non-existence of low-dimensional deterministic chaos, implying the possibility of "accurate" short-term prediction, and furthermore, gaining knowledge about the number of essential variables necessary for mathematical modelling of the structure of the rainfall dynamics. Thus, the objective of this case study is twofold. Firstly, to review and address some of the important issues, such as the influence of the temporal correlations, in the application of chaos identification methods in rainfall, especially focused on the estimation of the geometric and dynamic invariants such as correlation dimension and Lyapunov exponents. Secondly, and more important, a question is posed regarding the existence of structurally different chaotic dynamics in the rainfall using different temporal scales of the observables. This issue was initially addressed by Rodriguez-Iturbe et al. (1989), but never further investigated.

In this case study we review the previous studies investigating the existence of deterministic chaos in rainfall, the approaches applied and some of the limitations of the chaos diagnostic tools addressed. Furthermore we contribute with a new analysis of 15min, hourly, daily and weekly rainfall data from De Bilt meteorological station in the Netherlands. Wavelet-based transformation of the rainfall time series is used to produce an adequate stochastic surrogate for distinguishing between stochastic and possibly chaotic dynamics. The singular value decomposition method (referred to in Section 3.3.1) together with the BDS test to the residuals are used to investigate the presence of nonlinearity in the rainfall data. Interpretation of the results in the last part of the application lead to the general discussion and conclusion concerning the question of the existence of structurally different chaotic dynamics in the rainfall at different temporal scales.

### *6.3.2 Chaos in rainfall: a review of related work*

The possible existence of chaos in rainfall was first investigated by Hense (1987), who applied the correlation dimension method to a series of daily rainfall ($N$=1080 samples) recorded in Nauru Island. The existence of chaos in rainfall time series was indicated by presence of low-dimensional attractor with correlation dimension between $d_c$=2.4÷4.0. However, it is questionable that the length of the rainfall record was long enough to justify the correlation dimension obtained. Rodriguez-Itrube *et al.* (1989) investigated the existence of chaos in rainfall using the correlation dimension method and Lyapunov exponents. They analysed a rainfall record of $N$=1990 values, measured with a highly sensitive rain gauge with a sampling frequency of 8 Hz and then aggregated at equally spaced intervals of 15 s, from a single storm event in October 1980 in Boston. Estimation of a finite low-correlation dimension of about 3.7 provided preliminary evidence for the existence of chaos in storm rainfall data. It is interesting to mention that the analysed

rainfall record does not contain any zero values. The presence of chaos was further supported by the existence of a positive Lyapunov exponent.

Further evidence of the presence of chaos in storm rainfall was presented by Sharifi *et al.* (1990), who applied the correlation dimension method to examine data from three storms. The total number of samples of the time series, representing the time to 0.01 mm of rain, for each of the three storms were N=4000, 3991, and 3316 and the estimated correlation dimensions were $d_c$=3.35, 3.8, and 3.6, respectively. The study confirmed the results obtained by Rodriguez-Itrube *et al.* (1989). Tsonis *et al.* (1990) further investigated data representing the time between successive rainfall signals each corresponding to a 0.01mm of rain. The existence of structure with a low-dimensional attractor in this time series ($d_c$=2.4) indicated the possible existence of chaos. Islam *et al.* (1993) analysed the simulated rainfall intensity data using the correlation dimension method. From a data set of $N$=7200 samples, generated at 10 sec time step from a three-dimensional atmospheric model, they obtained a value for the correlation dimension of about $d_c$=1.5. Jayawardena and Lai (1994) investigated daily rainfall data from three rainfall stations in Hong Kong. The correlation dimension method, the Lyapunov exponents method, the Kolmogorov entropy method, and the nonlinear prediction method were used on the daily rainfall datasets with $N$=4015. Their study provided evidence of the existence of chaos in the daily rainfall data. The estimated correlation dimensions for the three stations are $d_c$=0.95, 1.76 and 1.65, respectively. They further reported a low predictive possibility for the daily rainfall in Hong Kong. Georgakakos *et al.* (1995) analysed data from 11 storm events in Iowa City, and reported the possible existence of chaos. The correlation dimensions were found to range from $d_c$=2.8 to $d_c$=7.9 in the high-intensity scaling region, while in the low-intensity scaling region they ranged from 0.6 to 1.6. Finally, Sivakumar *et al.* (1999) investigated the daily rainfall data of different record lengths (max $N$=10958) observed at six rainfall stations in Singapore. They reported correlation dimensions between $d_c$=1.01÷1.06 for the six stations respectively.

The discrepancy in the attractor dimensions of the rainfall dynamics resulting from the above mentioned work could be caused by some of the drawbacks of the correlation dimension method; see Section 3.3.2 for discussion. Closer examination by several authors (see Sivakumar, 2000 for an overview) showed that the straightforward application of the correlation dimension method suffers from several problems, such as the number of the points needed for a reliable estimation of the correlation dimension, the choice and sensitivity of appropriate time delay for the reconstruction of the phase space and the effect of noise. However, one of the most important considerations is the existence of *temporal correlations* (see Figure 3.20b in Section 3.3.2), which was not properly addressed by the authors seeking a low-dimensional attractor in the rainfall. Another very important issue for the analysis of the rainfall time series is the existence of large number of zeros, which basically requires very long time series of rainfall records in order to populate the reconstructed phase-space of the system. These two factors are introducing an upward bias in the estimation of the correlation dimension from its definition; see Equation 3.59.

As already presented in Section 3.3.2, the problem of the temporal correlations, which is more pronounced for frequently sampled time series data, is solved by Provenzale *et al.*, (1992) and is discussed in Velickov (2001). For illustration, we present here the effect of temporal correlations on the correlation dimension estimation for the Lorenz system,

discussed in Section 3.1, Example 3.1. Figure 6.3.2 shows the correlation sum for the Lorenz data.

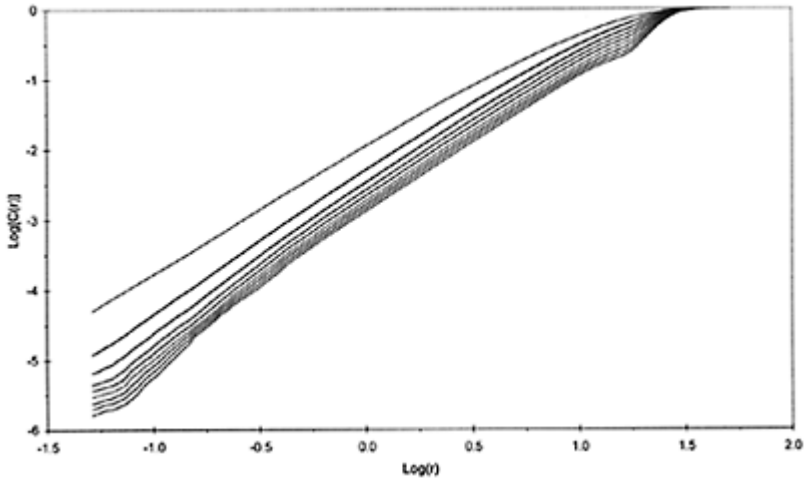



Figure 6.3.2. Correlation integral (sum) for the Lorenz system (Example 3.1). A double logarithmic plot was chosen for better visual presentation of the power law scaling between the correlation sum *C(r)* and the length scales *r*. The correlation sum was computed for different embedding dimensions *m*=2–10 without accounting for the temporal correlations (the most upper line represents *m*=2).

The relationship between the correlation exponent (slope) and embedding dimension *m* for different scaling regions *r* for the Lorenz data is presented in Figure 6.3.3.

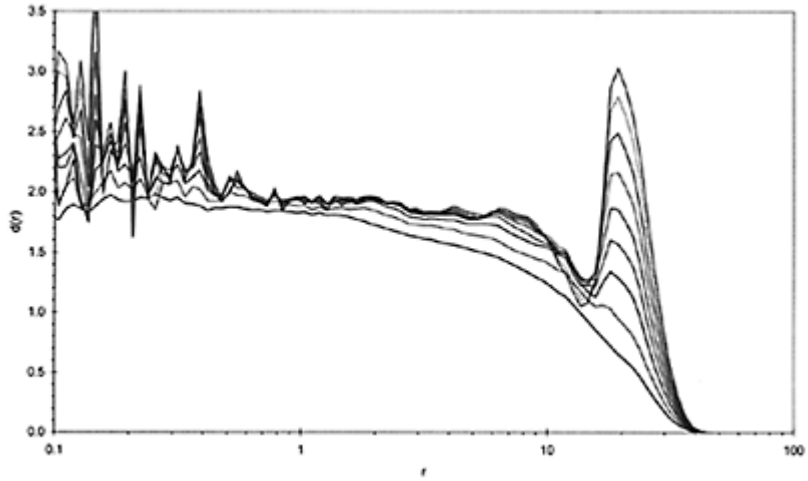



Figure 6.3.3. Relationship between the correlation exponent *d(r)* and the scaling regions *r* for different embedding dimension *m*. This relationship suggests lower "plateau" for the correlation dimension (about 1.8–1.9) when temporal correlations are not accounted for. The exact correlation dimension for the Lorenz attractor is 2.06 (Grassberger and Procaccia, 1983). The present figure shows also sensitive correlation exponents for the lower scaling regions. The lowest curve corresponds to $m$=2.

The space-time separation plot, which indicates the number of pairs as a function of two variables, the time separation $\Delta t$ and the distance $r$, calculated for the Lorenz system is presented in Figure 6.3.4. Technically, the plot shows the contour lines for 10%, 20%, 30%…, of the pairs with a given temporal separation $\Delta t$. In other words, the contour lines indicate the distance we have to go to find a given fraction of pairs, depending on their temporal separation. Only for values of $\Delta t$ where the contour lines are becoming flatter, does the temporal correlation not bias the correlation sum.

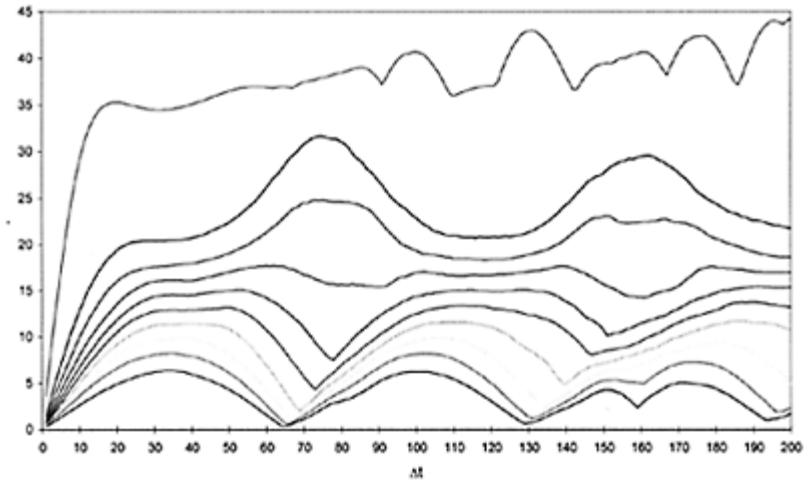



Figure 6.3.4. Space-time separation plot for the Lorenz data. Contour lines are shown at the spatial distance $r$ where for a given temporal separation $\Delta t$ a fraction of 10%, 20%, 30%,…(lines from below) of pairs are found. The saturation for all contour lines is reached above $\Delta t$=30 time steps.

The curves shown in Figure 6.3.4 suggest that in the estimation of the correlation dimension (Equation 3.61) we must consider a time window of about 30 time steps. If we do not discard at least those pairs, which are less than 30 time steps apart, then we obtain the correlation exponents shown in Figure 6.3.3. The relationship between the correlation exponent (slope) and embedding dimension $m$ for the Lorenz system when the temporal correlations are accounted for is presented in Figure 6.3.5.

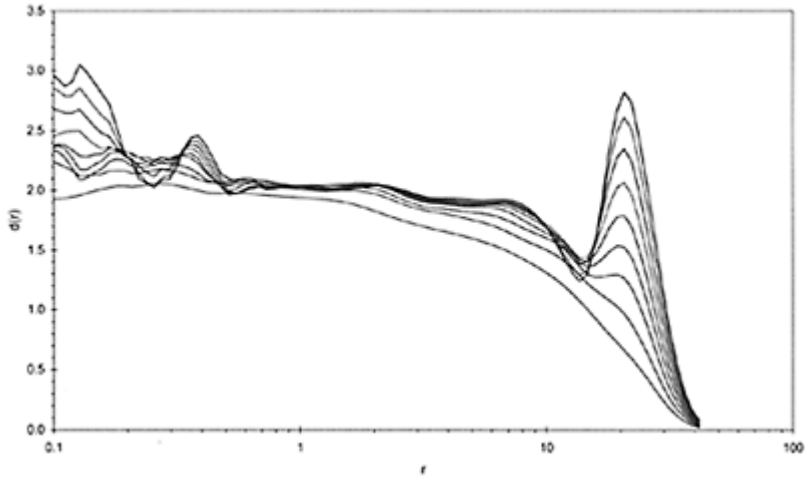



Figure 6.3.5. Relationship between the correlation exponent *d(r)* and embedding dimension *m* for different scaling regions *r* for the Lorenz data. This relationship suggest "plateau" for the correlation dimension (about 2.0–2.08) when temporal correlations are eliminated.

The Lorenz data presented in the above experiment is considered to be noise free. In order to illustrate the sensitivity of the correlation dimension on the presence of noise in the data (rainfall data in general is considered as highly noisy data), the correlation dimension estimation was carried out on the Lorenz data polluted with 5% of zero mean white noise. Figure 6.3.6 shows the correlation exponents for different embedding dimensions at different scaling regions.

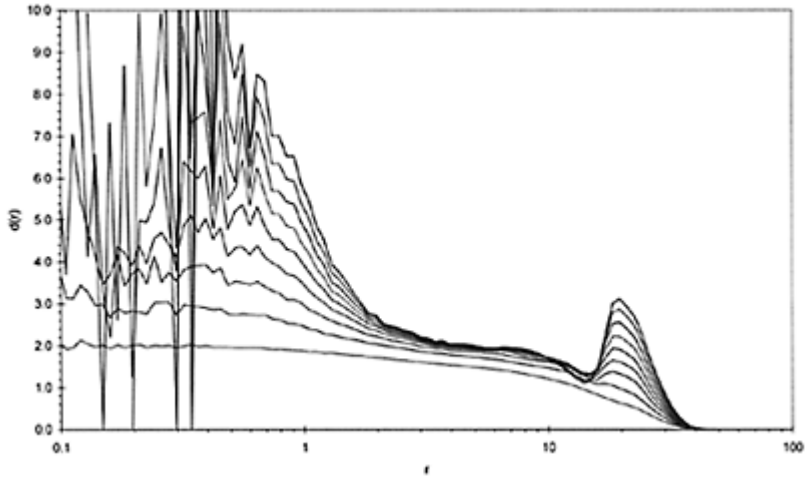



Figure 6.3.6. Relationship between the correlation exponent *d(r)* and embedding dimension *m* for different scaling regions *r* for the Lorenz data polluted with 5% white noise.

It is obvious that the presence of noise seriously affects the local correlation exponents at the smaller scaling regions. Since the values of the rainfall, especially for frequently sampled records, can be comparable to the errors (noise) of the measuring devices, several authors (e.g. Schreiber, 1993) advocate the application of the noise reduction algorithms before applying the correlation dimension analysis to the rainfall data.

### *6.3.3 Analysis of 15min, hourly, daily and weekly rainfall time series*

#### STUDY AREA AND DATA USED

In the present application, the methods and techniques elaborated in Chapter 3 were applied to analyse rainfall data from De Bilt meteo station in the Netherlands. As mentioned above, the main objective of this analysis is the quest for the existence of structurally different chaotic dynamics in the rainfall using different temporal scales of the observables. This important issue, which may reveal the differences in the low-dimensional attractors that were obtained for the rainfall dynamics, has not been investigated in the previous studies. The rainfall depths are recorded with a continuous tipping bucket rain gauge capable of aggregating the rainfall on different time intervals. The rainfall data were provided by the KNMI and are available for the period of 44 years between 1955 and 1998 in a form of complete 15min, hourly, daily and weekly time series. The length of the 15min rainfall records is $N$=1542816 data points, for the hourly

rainfall records there are *N*=385440 data points, for the daily records there are *N*=16071, and for the weekly records there are N=2296 data points.

## ANALYSIS OF THE 15MIN RAINFALL DATA

In order to investigate the existence in of chaotic behaviour in the rainfall dynamics on different temporal scales, we first analysed the 15min rainfall time series. Figure 6.3.7 shows the variation of the 15min rainfall data.

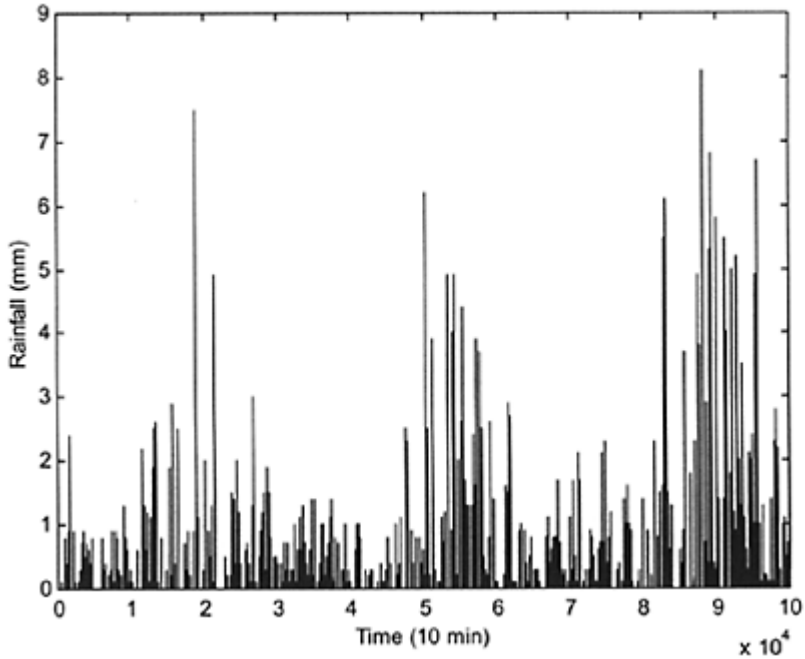



Figure 6.3.7. Variation of the 15min rainfall data. First 100000 samples are shown.

It is interesting to note that out of the total number of samples for the 15min rainfall time series only about 7% are non-zeros. In order to compare the results from the analysis we generated a data sequence of times between rainfall depth>0.1mm. Figure 6.3.8 shows a part of this sequence.

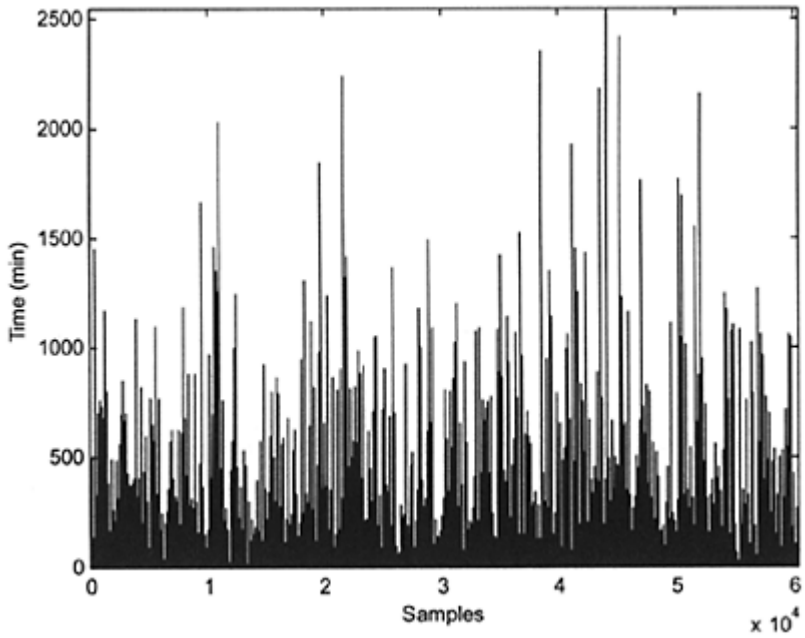



Figure 6.3.8. Sequence of times between rainfall depth>0.1mm.

The time delay was computed using the autocorrelation function and the average mutual information, explained in Section 3.3.3. Figure 6.3.9 shows both functions for different time lags.

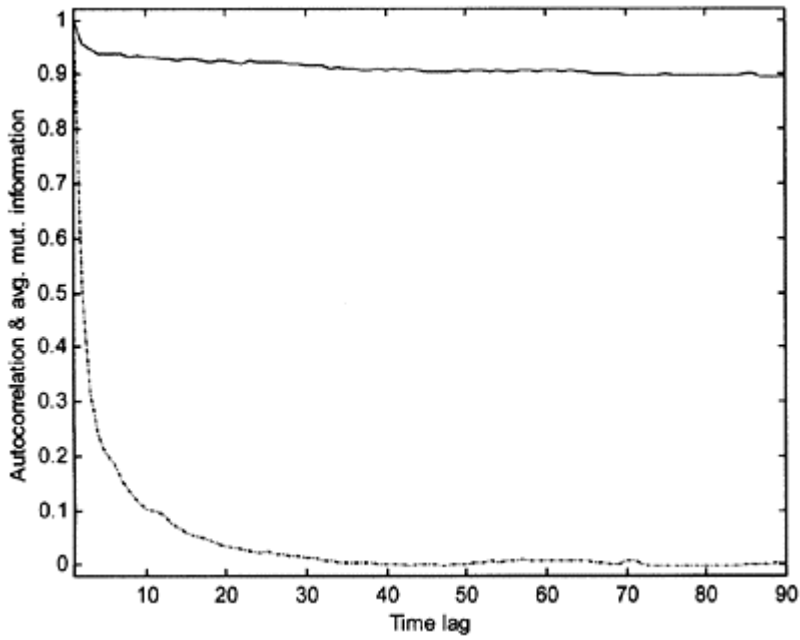



Figure 6.3.9. Autocorrelation function (dash-dotted line) and the normalized average mutual information (solid line) as a function of the time lags for the 15min rainfall time series. The time lag is measured in 15 min units.

The first zero-crossing of the autocorrelation function indicates time delay of about 42–48 time lags. The average mutual information reaches the first minimum at about 22–24 time lags. The correlation sum and the correlation exponent were computed using the Equation 3.61 as explained earlier. Figure 6.3.10 shows the correlation sum for the 15min rainfall data.

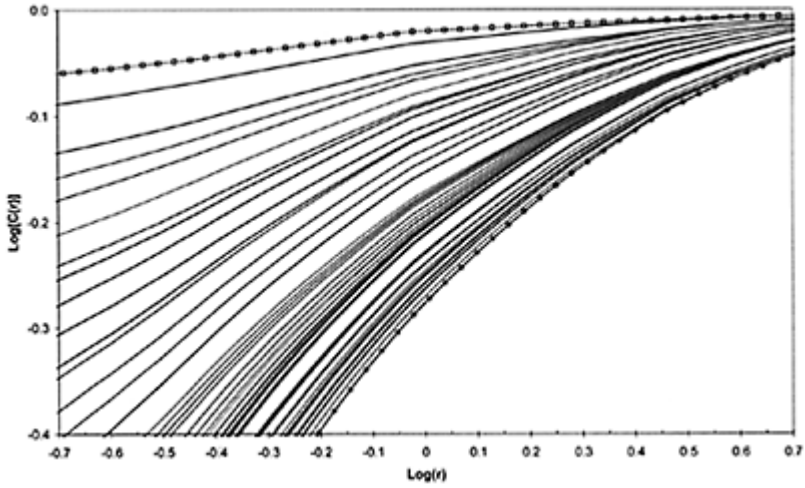



Figure 6.3.10. Correlation sum of the 15min rainfall time series for different embedding dimensions, between $m$=2 (squares) and $m$=50 (circles) The time delay used to produce this figure is $\tau$=48 time samples.

The sensitivity of the average correlation exponent (correlation dimension) on different time delays is presented in Figure 6.3.11.

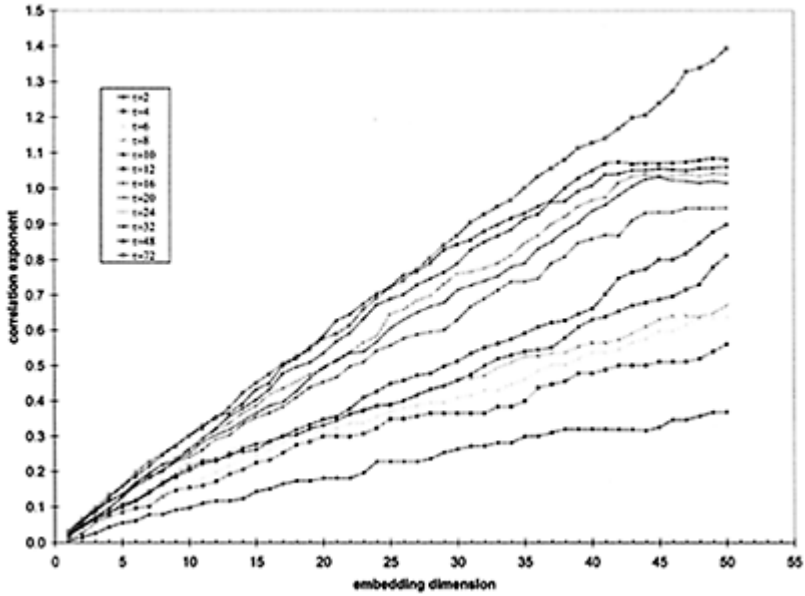



Figure 6.3.11. Relationship between the correlation exponent and the embedding dimension *m* for the 15min rainfall time series using different time delays τ.

The correlation exponent increases with an increase in the embedding dimension up to a certain value and further saturates (when using time delays between τ=24 and τ=48). The saturation value of the correlation exponent, that is the correlation dimension, is $d_c$ =1.05 (uncertainty 0.05) which indicates the presence of an attractor in the rainfall dynamics. Application of the Taken's embedding theorem suggests a dimension of the reconstructed phase-space (integer number) as $m=2d_c+1=3$. A view of the reconstructed phase space in three dimensions for the 15min rainfall time series is presented in Figure 6.3.12.

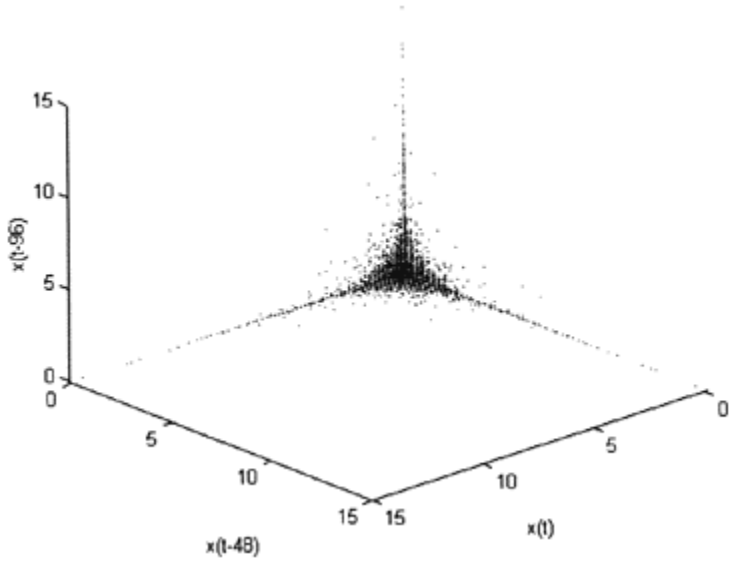



Figure 6.3.12. A view on the attractor of the 15min rainfall time series in three dimensions.

The estimation of the embedding dimension *m* was further checked using the FNN algorithm described in Section 3.3.2. Both embedding dimensions, estimated on the 15min rainfall time series and on the sequence of times between rainfall depth>0.1mm, indicate an embedding dimension *m*=3; see Figure 6.3.13.

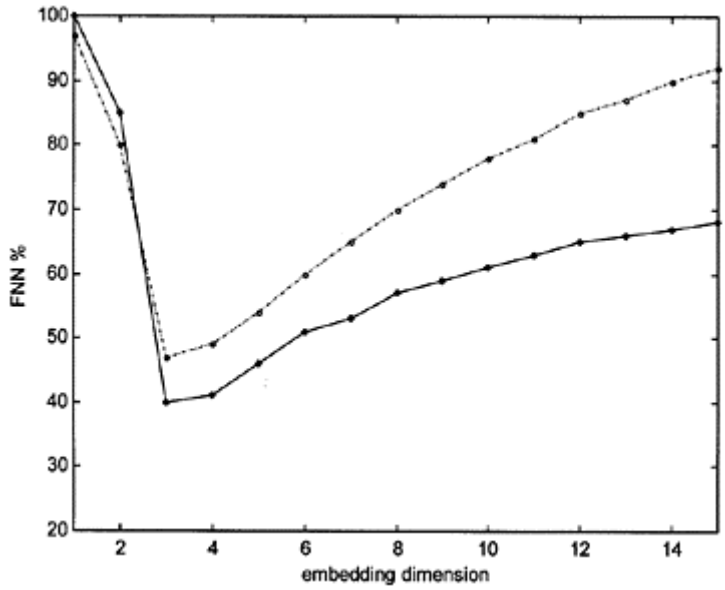



Figure 6.3.13. The percentage of the false nearest neighbours as a function of the embedding dimension for the 15min rainfall data (solid) and the times between rainfall>0.1mm (dash-dotted).

The saturation value of the correlation exponents (i.e. correlation dimension), as presented in Figure 6.3.11, occurs at high embedding dimension $m$=40. This value of the embedding dimension at which the saturation of the correlation dimension occurs is considered to provide the upper bound of the phase-space sufficient to fully describe the dynamics of the attractor. Furthermore, according to the theory of nonlinear dynamics, the embedding dimension of the phase-space is equal to the number of variables present in the evolution of the system dynamics. Therefore, the results from the analysis of the 15min rainfall time series indicate the existence of low-dimensional attractor that can be modelled with the minimum number of *essential variables* equal to 3 and the number of *sufficient variables* equal to 40. Such a low number of the essential variables implies the existence of persistency in the rainfall dynamics over very small temporal scales.

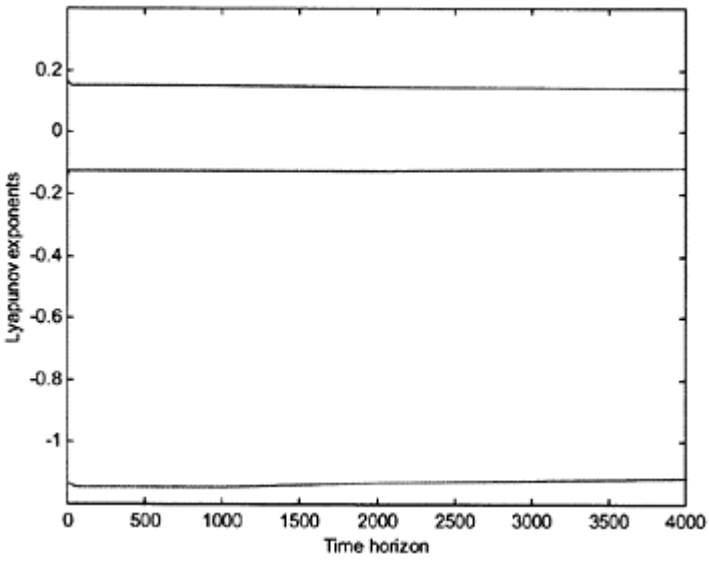



Figure 6.3.14. Lyapunov exponents for the 15min rainfall data.

The Lyapunov exponents estimated from the 15min rainfall time series, using the methodology described in Section 3.3.4, are presented in Figure 6.3.14. The largest Lyapunov exponent is estimated as $\lambda_1$=0.18 (uncertainty 0.02) which indicates the presence of a divergence in the nearby orbits in the reconstructed phase-space during the dynamical evolution of the system, and thus a loss of predictive capabilities. The sum of the Lyapunov exponents is negative, thus indicating presence of dissipation mechanisms in the rainfall dynamics and the existence of chaotic dynamics.

## ANALYSIS OF THE HOURLY RAINFALL DATA

A similar analysis was carried out using the hourly rainfall time series. Figure 6.3.15 shows the variation of the hourly rainfall data and Figure 6.3.16 shows the running variance for the hourly rainfall data computed on 72 hours (3 days) temporal window.

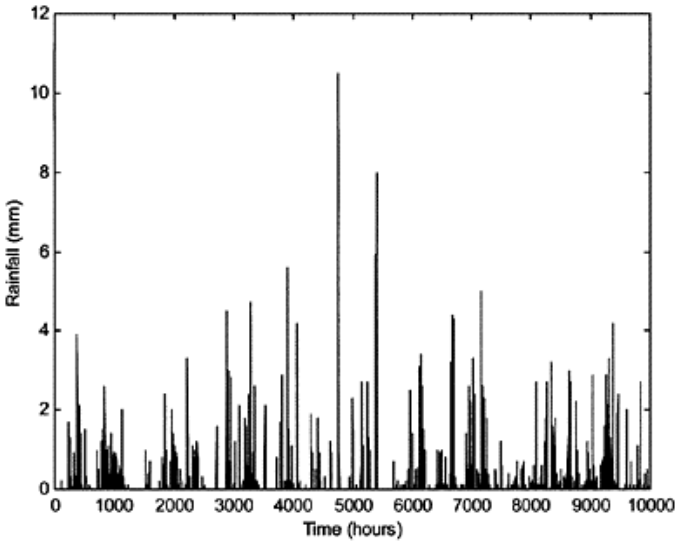



Figure 6.3.15. Variation of the hourly rainfall. First 10000 samples are shown.

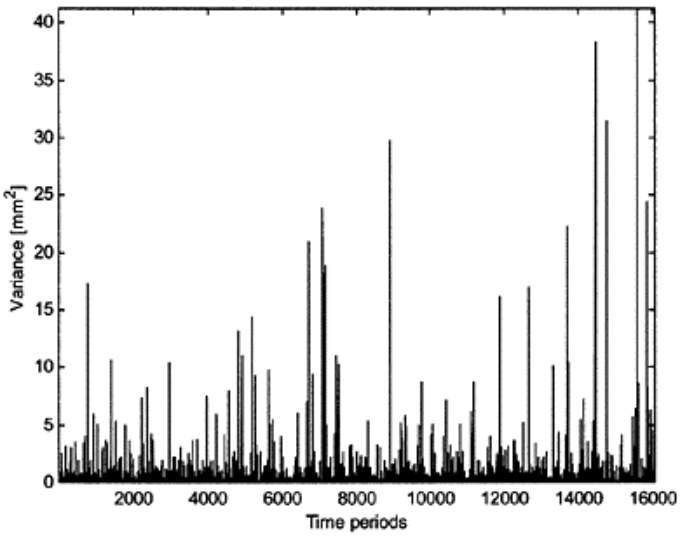



Figure 6.3.16. Running variance for the complete hourly rainfall time series. The time window for

calculation of the variance was set to 72 hours (3 days).

The hourly time series of the rainfall data contains about 12% of non-zero values, which still indicates the presence of a very large portion of zero values in the computation of the correlation dimension. In order to check the computation of the correlation dimension, we generated a time series of the rainfall differences (i.e. intensity), and further applied the noise reduction algorithm on the hourly data described in Section 3.3.6. Furthermore, a stochastic surrogate data set to the hourly rainfall time series was analysed in order to distinguish between possible stochastic and chaotic dynamics. This surrogate data set was created based on the continuous wavelet transformation (see Section 3.3.7, Equation 3.80) in order to preserve the power spectrum of the original time series. During the inverse transformation, the times (phases) of the wavelet coefficients were randomised in order to generate a stochastic surrogate. The autocorrelation and the average mutual information functions for the hourly time series data are presented in Figure 6.3.17.

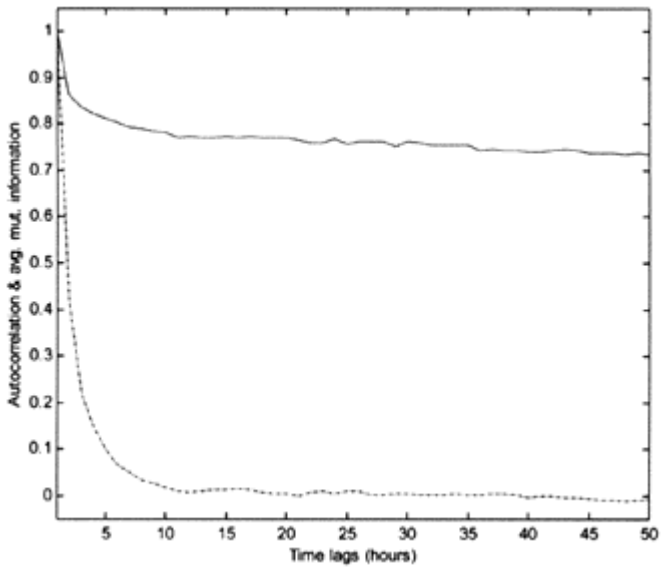



Figure 6.3.17. Autocorrelation and normalised average mutual information for the hourly rainfall time series.

The autocorrelation function indicates optimal time delay of about 22 hours (first zero), whereas the average mutual information indicates optimal time delay of 12 hours (the first minimum). Figure 6.3.18 shows the correlation sum for the hourly rainfall data.

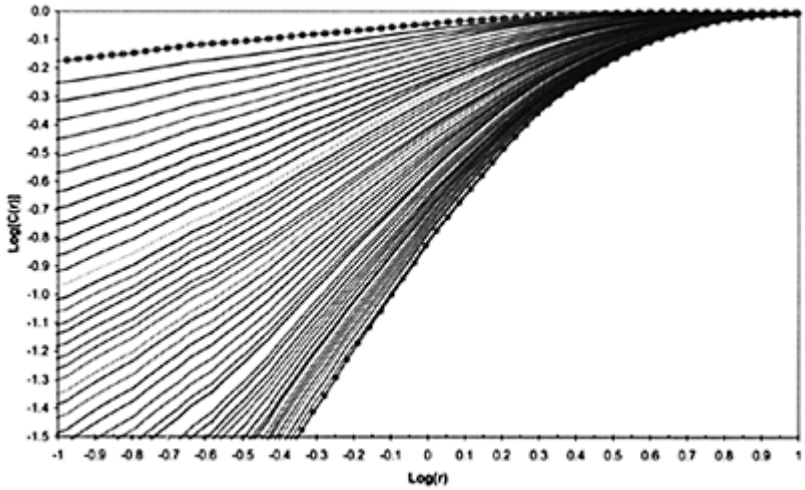



Figure 6.3.18. Correlation sum of the hourly rainfall time series for different embedding dimensions, between *m*=2 (squares) and m=50 (circles). The time delay used to produce this figure is $\tau$=12 hours.

The temporal window used in the computation of the correlation sum was set to $\Delta t$=6 hours based on the space-time separation plot for the hourly data, presented in Figure 6.3.19.

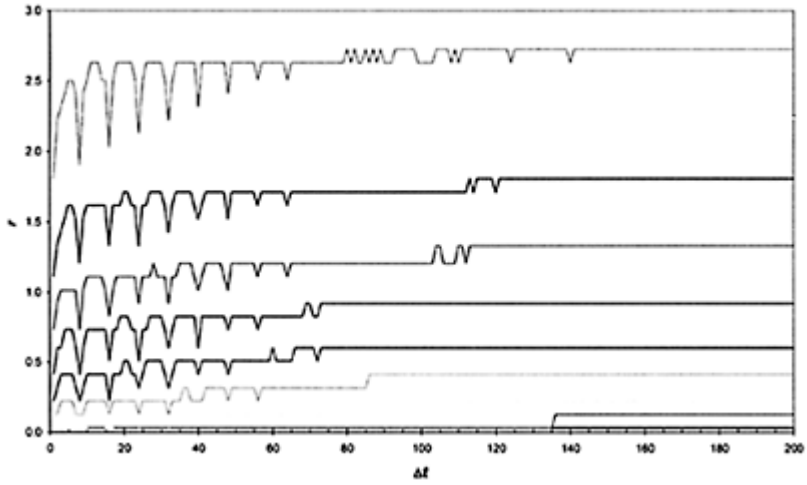



Figure 6.3.19. Space-time separation plot for the hourly rainfall data. Contour lines are shown at the spatial distance $r$ where for a given temporal separation $\Delta t$ a fraction of 10%, 20%, 30%,…(lines from below) of pairs are found. The first saturation for all contour lines is reached above $\Delta t$=6 time steps (hours). Most of the contour lines become flat after $\Delta t$=72 time steps (3 days).

The relationship between the average correlation exponent (correlation dimension) and the different time delays employed for the reconstruction of the phase-space based on the hourly rainfall data is presented in Figure 6.3.20.

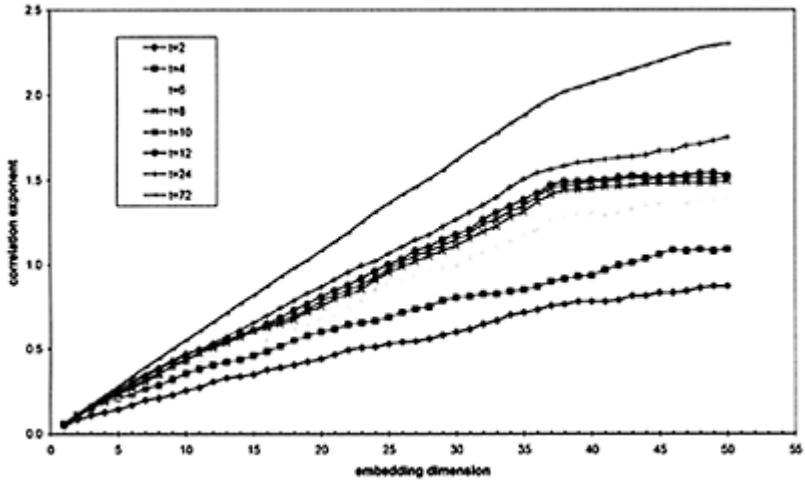



Figure 6.3.20. Relationship between the correlation exponent and the embedding dimension *m* for the hourly rainfall time series using different time delays τ.

The correlation exponent increases with an increase of the embedding dimension up to a certain value and further saturates (when using time delays between τ=8 and τ=24 hours).

The saturation value of the correlation exponent, that is the correlation dimension, is $d_c$=1.52 (uncertainty 0.1) which indicates the presence of an attractor in the rainfall dynamics. Application of Taken's embedding theorem suggests a dimension of the reconstructed phase-space (integer number) as $m=2d_c+1=4$ or 5. The false nearest neighbourhood method together with the Lyapunov spectrum (and dimension), see Figure 6.3.26 below, indicate an optimal embedding dimension of $m$=4 for the hourly rainfall dynamics. A view (projection) of the reconstructed phase space in three dimensions for the hourly rainfall data is presented in Figure 6.3.21.

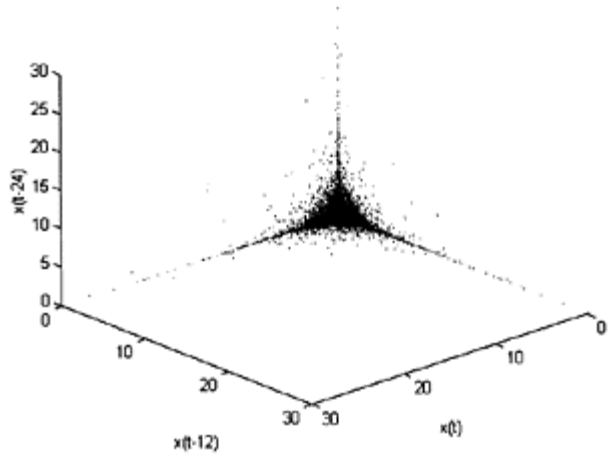



Figure 6.3.21. A view (projection) of the attractor of the hourly rainfall dynamics in three dimensions.

Figure 6.3.20 shows that the saturation value of the correlation dimension occurs at embedding dimension $m$=38. The results from the analysis of the hourly rainfall time series for De Bilt station indicate the existence of a low-dimensional attractor that can be modelled with the minimum number of *essential variables* equal to 4 and the number of *sufficient variables* equal to 38. It should be noted, however, that the correlation dimension analysis provides information only on the number of variables influencing the dynamics of the system, and does not identify the variables for the mathematical model of the rainfall dynamics. The effect of the removal of the temporal correlations, by using the temporal separation window, is illustrated in Figure 6.3.22.

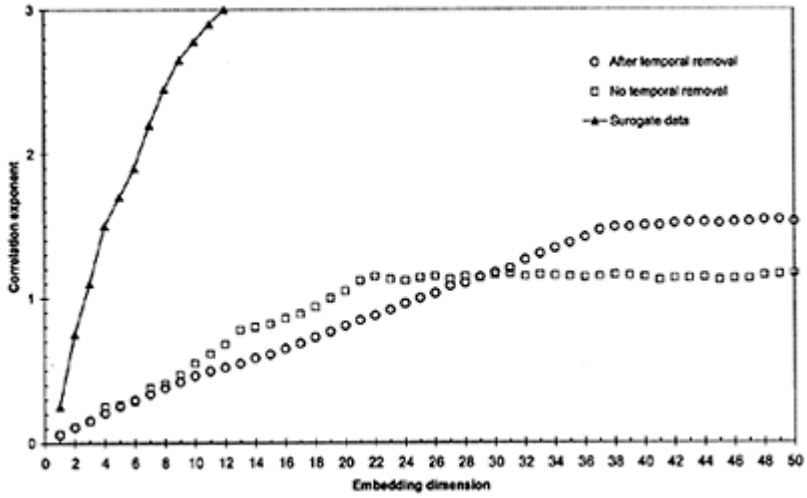



Figure 6.3.22. The effect of the temporal correlations on the dimension estimation. No removal of the temporal correlations contributes to underestimation of the correlation dimension.

The correlation dimension estimation for the hourly rainfall data was carried out using the difference rainfall data (intensities), “noise-free” rainfall data and the surrogate rainfall data. The relationship between the correlation exponent for different embedding dimensions for the three time series is presented in Figure 6.3.23.

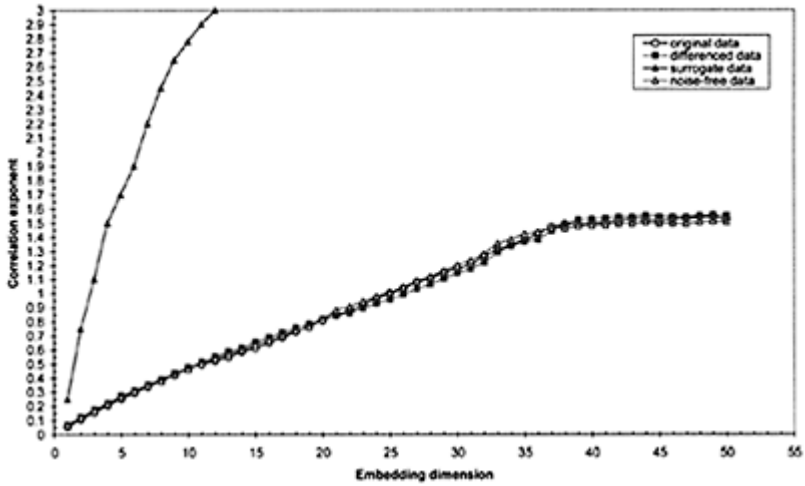



Figure 6.3.23. Relationship between the correlation exponent and the embedding dimension *m* for the difference rainfall data, "noise-free" data and the surrogate data. The time delay $\tau$=12 (12 hours).

Both, the "noise-free" data and the difference data indicate similar saturation values for the correlation dimension (between 1.48–1.56), which are not significantly different with the correlation dimension estimated from the original time series. The correlation exponent for the stochastic surrogate data does not show saturation and constantly increases with the increase of the embedding dimension. This indicates that the rainfall dynamics is different from a random process.

In order to investigate the nonlinearity in the rainfall data, the singular value decomposition technique, explained in Section 3.3.1, was used to extract the linearly independent principal components. Figure 6.3.24 shows the results of the analysis.

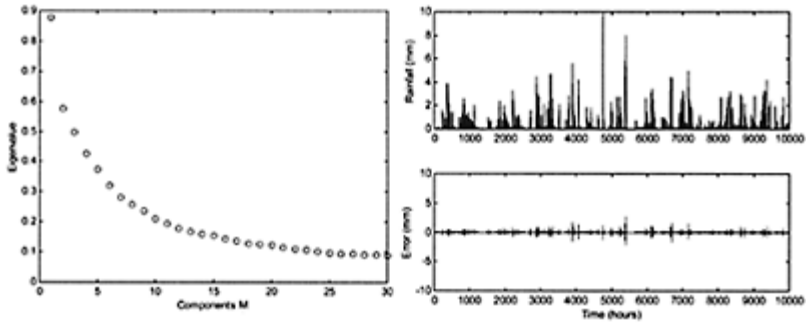



Figure 6.3.24. (a) The eigenvalue spectrum of the covariance matrix of the hourly rainfall data. First 25 eigenvalues and the corresponding eigenvectors were used to reconstruct the hourly precipitation; (b) Original (grey line) and reconstructed (line) time series of the hourly rainfall with the errors.

The differences between the two time series were further analysed statistically. Looking at the residuals, they appear substantially uncorrelated according to the behaviour of their autocorrelations, but when applying some transformation functions on the residuals, namely absolute and squared residuals, some clear signals of autocorrelations are found (see Figure 6.3.25). This basically denotes a lack of independence between the residuals and some form of higher-order dependence in the original data.

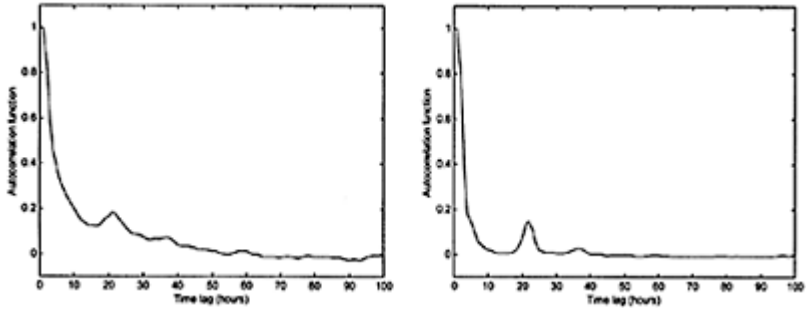



Figure 6.3.25. Autocorrelation function for the: (a) absolute and (b) squared residuals.

In order to confirm the presence of nonlinearity in the rainfall data, we applied the BDS test (Brock *et al.*, 1987) to the residuals. This test is based on the correlation dimension analysis. The *BSD* statistics is written in the form:

$$BDS = \sqrt{N}\left[C(m,r) - C(1,r)^m\right] / \sqrt{V} \tag{6.3}$$

where $m$ is the embedding dimension, $r$ is the scaling region, $C(m,r)$ is the correlation sum and N is the number of data points used to calculate the correlation sum. Brock *et al.* (1987) showed that, under the hypothesis of independence and identical distribution, the *BDS* statistics is asymptotically the normal standard distribution. The calculation of the *BDS* statistics using different values for $m$ and $r$, as shown in Figure 6.3.18, resulted in values between 16.6–98.2. These values reject the null hypothesis of independence and an identical distribution of the residuals and, consequently, the linearity hypothesis on the rainfall data.

Finally, the Lyapunov exponents estimated from the hourly rainfall time series, are presented in Figure 6.3.26. There is a presence of positive Lyapunov exponent $\lambda_1$=2.1 (uncertainty 0.1) which indicates a loss of information of 2.1 bits/hour. The presence of two negative Lyapunov exponents indicate strong dissipation mechanisms in the rainfall dynamics and suggests a Lyapunov dimension of $d_\lambda$=3.87, which confirms the optimal embedding dimension $m$=4 for the hourly rainfall dynamics.

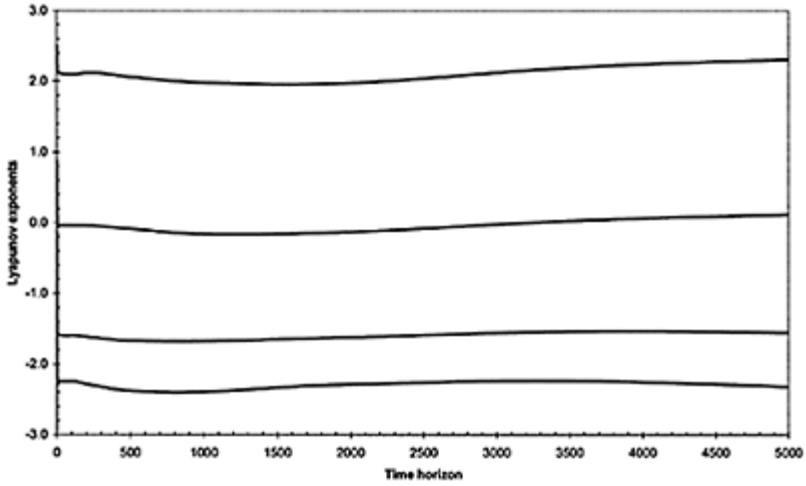



Figure 6.3.26. Lyapunov exponents for the hourly rainfall data.

ANALYSIS OF THE DAILY RAINFALL DATA

Figure 6.3.27 shows the variation of the daily rainfall data. The total number of samples is $N$=16071 with about 55% of non-zero values.

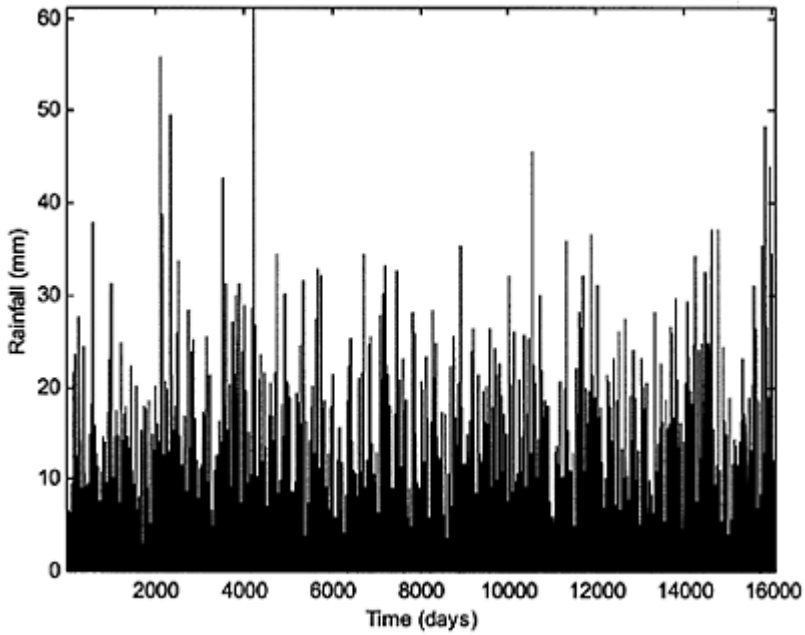



Figure 6.3.27. The complete time series of the daily rainfall data.

The autocorrelation and the average mutual information functions for the daily time series data are presented in Figure 6.3.28.

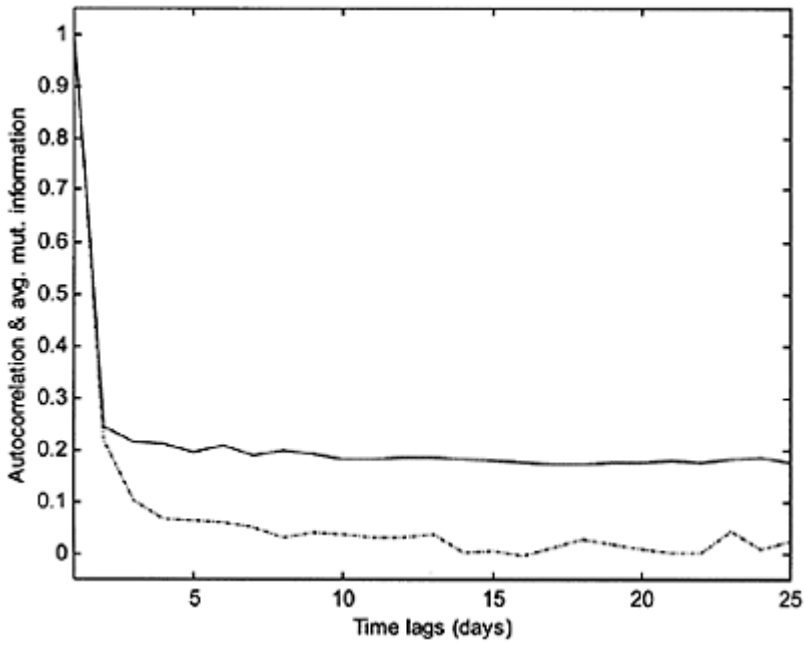



Figure 6.3.28. Autocorrelation and normalised average mutual information for the daily rainfall time series.

The autocorrelation function indicates optimal time delay of about 7–8 days (very close to zero), whereas the first zero-crossing is at 14 days. The average mutual information indicates an optimal time delay of 4–5 days (the first minimum). Figure 6.3.29 shows the correlation sum for the daily rainfall data.

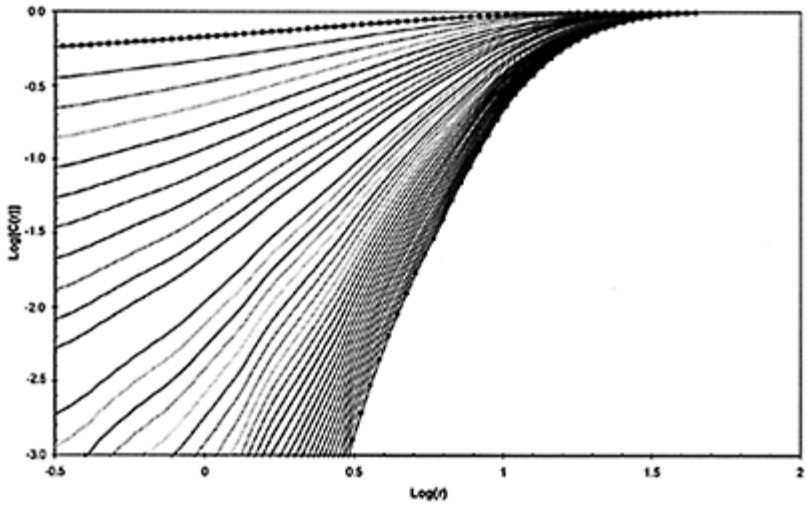



Figure 6.3.29. Correlation sum of the daily rainfall time series for different embedding dimensions, between $m$=2 (squares) and $m$=40 (circles). The time delay used to produce this figure is $\tau$=4 days.

The relationship between the average correlation exponent (correlation dimension) and the different time delays employed for the reconstruction of the phase-space based on the daily rainfall data are presented in Figure 6.3.30.

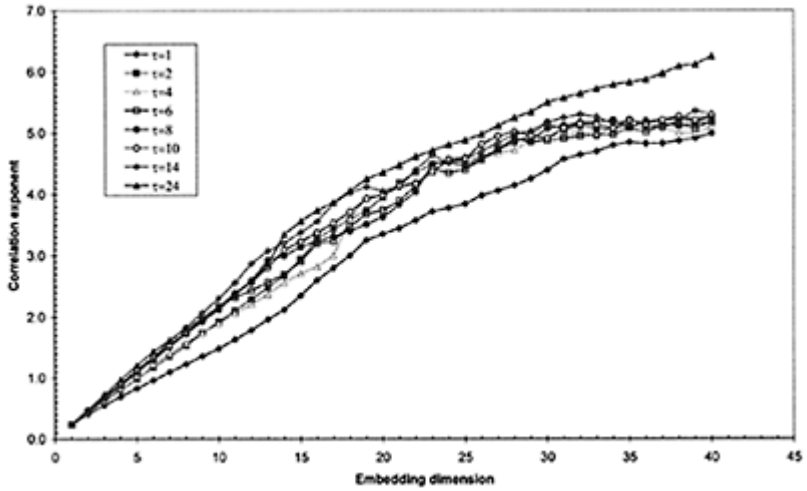



Figure 6.3.30. Relationship between the correlation exponent and the embedding dimension *m* for the daily rainfall time series using different time delays τ.

The correlation exponent increases with an increase of the embedding dimension up to a certain value and further saturates (when using time delays between τ=2 and τ=14 days). The saturation value of the correlation dimension for the optimal time delay of t=4 days, is $d_c$=5.12 (uncertainty 0.2) which indicates presence of an attractor in the daily rainfall dynamics. Application of the Taken's embedding theorem suggests a dimension of the reconstructed phase-space (integer number) as $m=2d_c+1=11$ or 12. Figure 6.3.30 further shows that the saturation value of the correlation dimension occurs at embedding dimension of about *m*=30. The results from the analysis of the daily rainfall time series for the De Bilt station indicate the existence of an attractor that can be modelled with the minimum number of *essential variables* equal to 11 and the number of *sufficient variables* equal to 30. The sensitivity of the correlation dimension analysis on the length of the time series was tested by using trial lengths of the daily time series, such as *N*=1000, *N*=2000, *N*=3000, *N*=5000, *N*=10000 and the complete record (*N*=16071). For the value of *N*>3000 the changes in the correlation dimensions were not significant, indicating that the available daily rainfall time series provides sufficient statistics for the calculation of the correlation dimension using higher trial embedding dimensions.

The correlation dimension estimation procedure was carried out further using the difference rainfall data (daily intensities), "noise-free" rainfall data and the surrogate rainfall data. The relationship between the correlation exponent for different embedding dimensions for the three time series is presented in Figure 6.3.31. The difference data shows a similar saturation value for the correlation dimension ($d_c$=5.08). Both correlation dimensions estimated on the original daily rainfall data and the differenced data stimulate an embedding dimension of *m*=12. However, the correlation dimension estimated on the

"noise-free" data as $d_c$=4.65, and the Lyapunov dimension (see Figure 6.3.32) indicate an optimal embedding dimension of $m$=11. The correlation exponent for the stochastic surrogate data does not show saturation and increases constantly with the increase of the embedding dimension. This indicates that the daily rainfall dynamics is different from a random process. The Lyapunov exponents estimated from the daily rainfall time series are presented in Figure 6.3.26. There are several positive Lyapunov exponents (maximum is $\lambda_1$=3.07, uncertainty 0.25) indicating hyper-chaotic dynamics. The sum of the Lyapunov exponents is negative $\Sigma\lambda_i$=−0.43, confirming the existence of an attractor in the daily rainfall dynamics. The Lyapunov dimension is $d_\lambda$=10.68, which suggests that the optimal embedding dimension is $m$=11 for the daily rainfall dynamics.

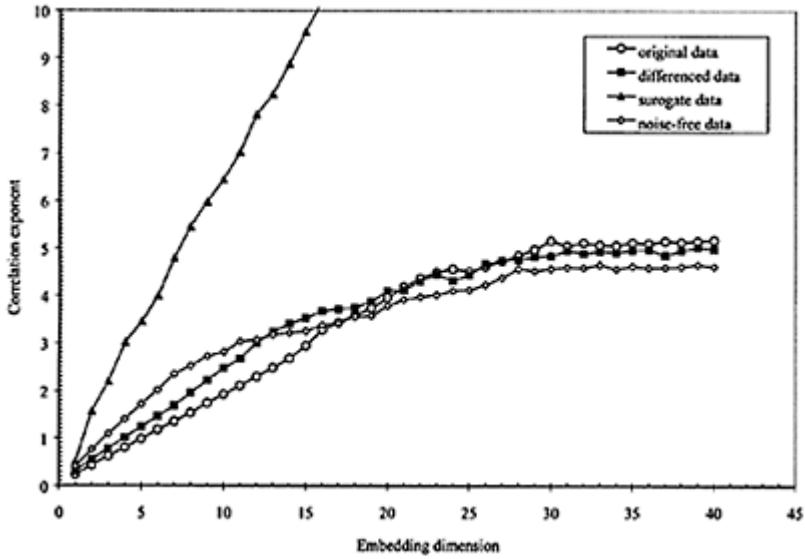



Figure 6.3.31. Relationship between the correlation exponent and the embedding dimension $m$ for the difference rainfall data, "noise-free" data and the surrogate data. The time delay used is $\tau$=4 days.

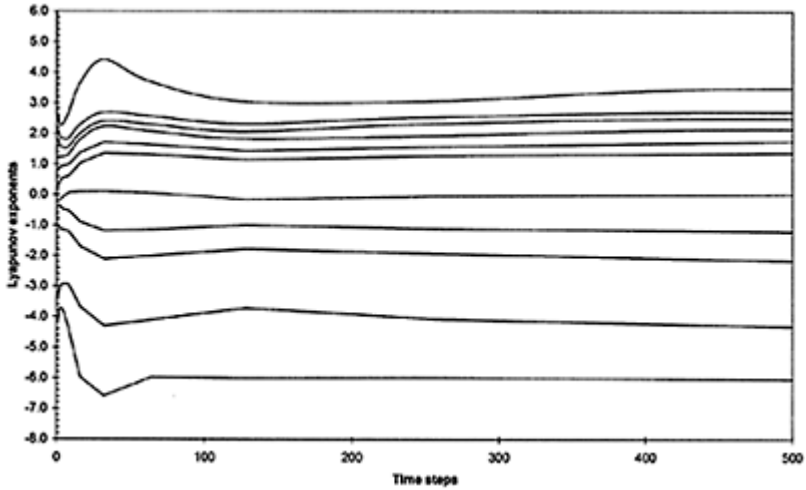



Figure 6.3.32. Lyapunov exponents for the daily rainfall data. Several positive exponents are present indicating hyper-chaos.

ANALYSIS OF THE WEEKLY RAINFALL DATA

The final analysis was performed using the time series of weekly rainfall data. Figure 6.3.33 shows the weekly rainfall data record. The total number of samples is $N$=2296 with about 94% of non-zero values.

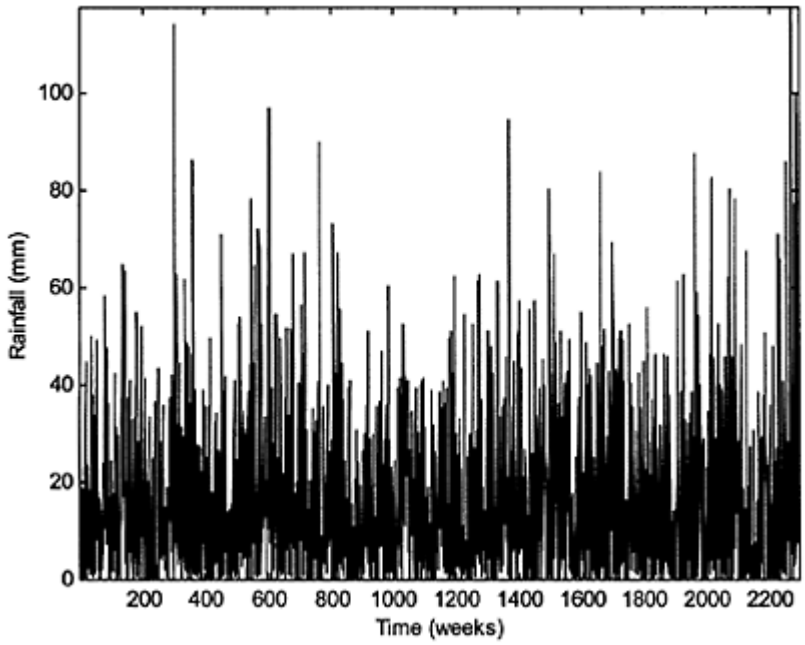



Figure 6.3.33. The complete time series of the weekly rainfall data.

The autocorrelation and the average mutual information functions for the daily time series data are presented in Figure 6.3.34.

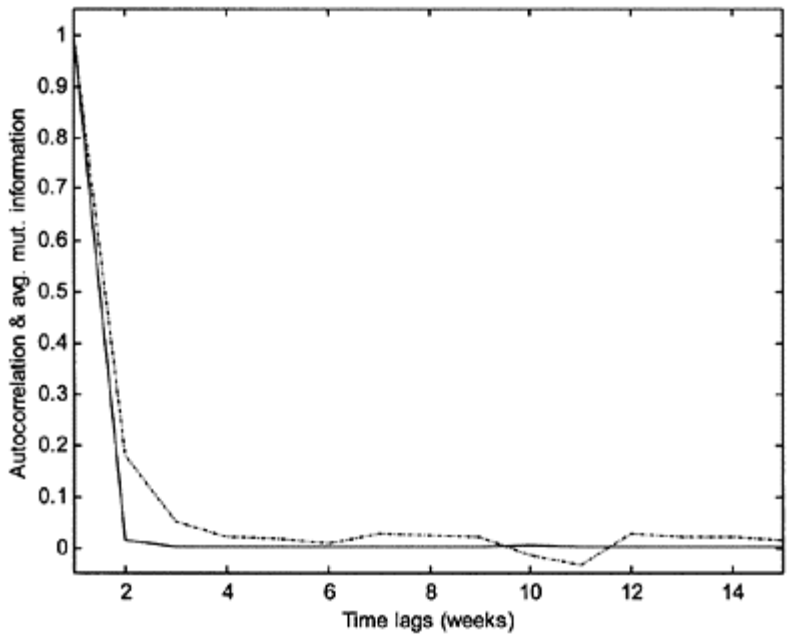



Figure 6.3.34. Autocorrelation and normalised average mutual information for the weekly rainfall time series.

The average mutual information function shows that there is a complete loss of information after 2 weeks indicating it is an optimal time delay. Figure 6.3.35 shows the correlation sum for the weekly rainfall data.

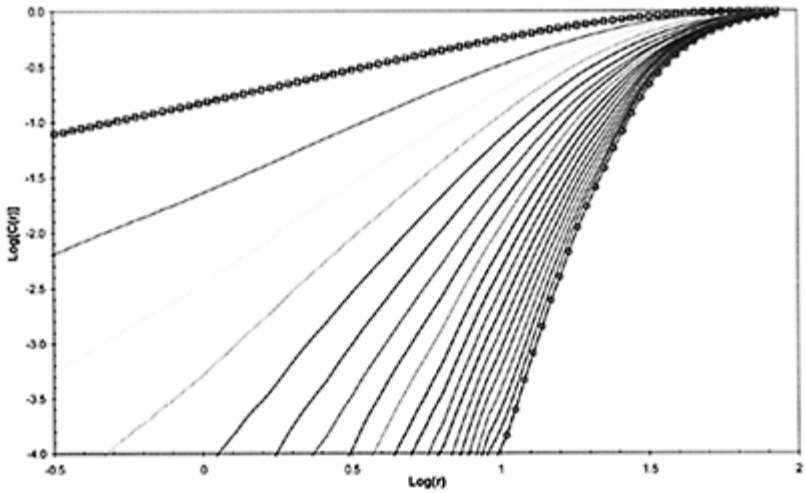



Figure 6.3.35. Correlation sum of the weekly rainfall time series for different embedding dimensions, between $m$=2 (squares) and $m$=20 (circles). The time delay used to produce this figure is $\tau$=2 weeks.

The relationship between the average correlation exponent (correlation dimension) and the different time delays employed for the reconstruction of the phase-space based on the weekly rainfall data is presented in Figure 6.3.36.

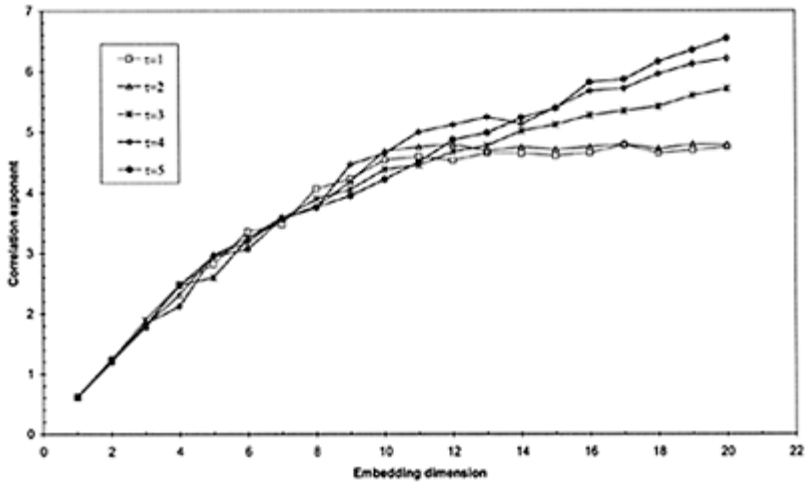



Figure 6.3.36. Relationship between the correlation exponent and the embedding dimension *m* for the weekly rainfall time series using different time delays τ.

The correlation exponent increases with an increase of the embedding dimension up to a certain value and further saturates (when using time delays of τ=1 and τ=2 weeks). The saturation value of the correlation dimension for the optimal time delay of τ=2 weeks, is $d_c$=4.71 (uncertainty 0.1) which indicates presence of an attractor in the weekly rainfall dynamics. Application of the Taken's embedding theorem suggests a dimension of the reconstructed phase-space (integer number) as $m=2d_c+1=11$. The saturation value of the correlation dimension occurs at an embedding dimension of about *m*=11 as well. The existence of a fractal correlation dimension (and thus attractor) for the reconstructed phase-space based on weekly rainfall data using time delays of τ=1 and τ=2 weeks supports the claims of some meteorologists (see e.g. Holton, 1992) that the global weather numerical models can be substantially improve for forecasting up to 14 days in the future. It is interesting to note that the analysis of the weekly rainfall data shows equal numbers for the essential variables and for the sufficient variables *m*=11 necessary to describe the dynamics. This implies that for the modelling and prediction of the weekly rainfall dynamics the number of the variables in the global weather models should be fewer than the number of the sufficient variables needed for the hourly and daily rainfall dynamics, 38 and 30 respectively.

In order to support these preliminary findings, we actually need a longer time series than 44 years (N=2296 samples). Due to the higher embedding during the computation of the correlation dimension, the number of points is substantially diminished. In this case, the data set of weekly rainfall of *N*=2296 points reduces to *N*=1354 after trial embedding to *m*=15 with a delay of τ=4 weeks. This may imply a shrinking of the region over which scaling exists, which is expressed through an appearance of "S"-type of curves in the

Log*[C(r)]*−Log*(r)* plot. However, the curves representing the correlation sums over different scaling regions presented in Figure 6.3.35 do now exhibit that behaviour, allowing for a good estimation of the correlation exponents (slopes).

The existence of an attractor in the weekly rainfall dynamics is further confirmed by the computation of the Lyapunov exponents; see Figure 6.3.37.

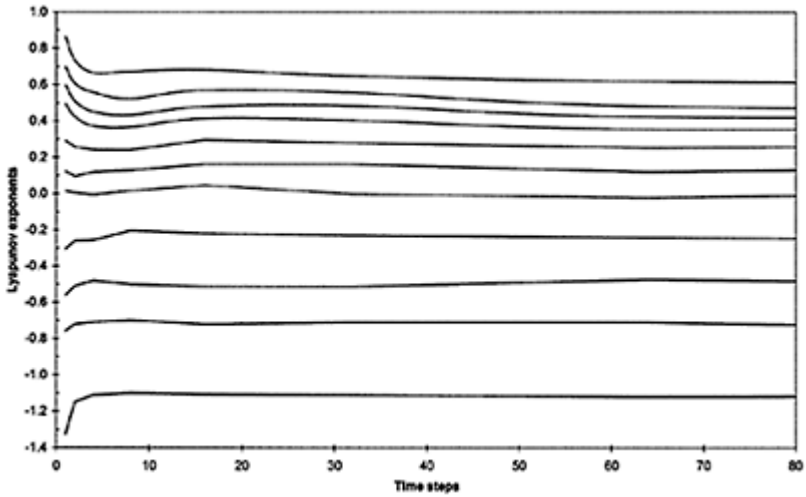



Figure 6.3.37. Lyapunov exponents for the weekly rainfall data.

There are several positive Lyapunov exponents (max is $\lambda_1$=0.67, uncertainty 0.1) indicating hyper-chaotic dynamics. The sum of the Lyapunov exponents is negative $\Sigma\lambda_i$=− 0.15, confirming the existence of an attractor in the weekly rainfall dynamics. The Lyapunov dimension is $d_\lambda$=10.92, which suggests that the optimal embedding dimension is $m$=11 for the weekly rainfall dynamics.

It is likely that at larger periods of rainfall aggregation, such as 2-weekly, 3-weekly or monthly rainfall, chaotic rainfall dynamics may not exist, and then at even larger periods of aggregation, such as seasonal and annual rainfall, a different type of chaotic dynamics is again established. Unfortunately, there are no time series of annual rainfall long enough to perform such a nonlinear time series analysis.

### *6.3.4 Predicting hourly and daily precipitation*

Based on the identified and reconstructed dynamics from the hourly and daily rainfall times series, an attempt was made to build forecasting models utilising the local modelling in phase space as elaborated earlier. Several types of univariate local models were constructed, namely, local linear, 2nd and 3rd order polynomial models. The local models were constructed from the first 2/3 of the available data, whereas the last 1/3 of the dataset was used to evaluate model performance by calculating the correlation coefficient between the measured and the predicted rainfall time series. The sensitivity of

these local models on the embedding dimension *m*, the time delay τ, and the number of neighbours *k* was investigated further. Figure 6.3.38 shows the relationship between the correlation coefficient and the number of neighbours in phase-space for different time delays for a prediction horizon of 1 hour ahead.

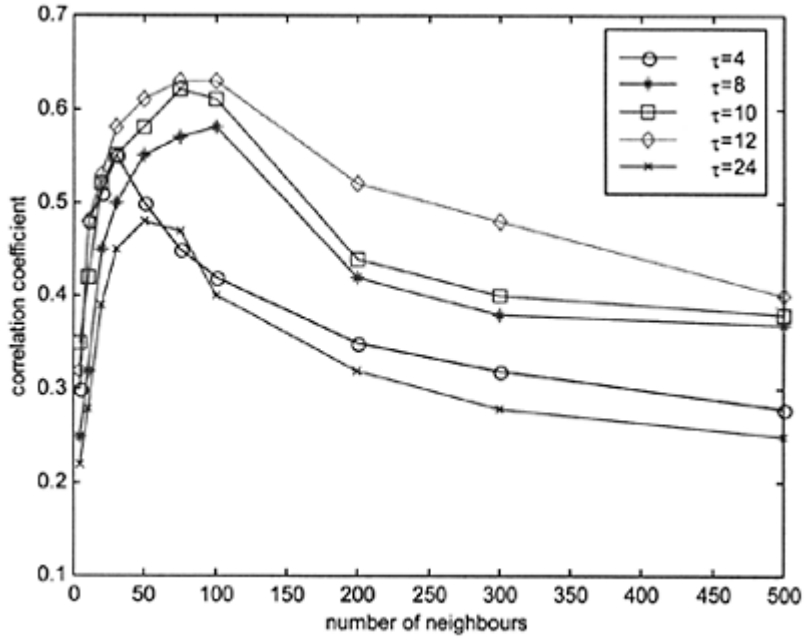



Figure 6.3.38. Correlation coefficient versus number of neighbours for different time delays. The prediction horizon is 1 hour ahead. Mixture of univariate local 3rd order polynomial models are used based on the reconstructed phase-space (*m*=4) from the hourly rainfall time series.

Figure 6.3.38 shows that the prediction accuracy increases with increasing the number of neighbours until a certain value and then decreases with a further increase in the number of neighbours. The optimal number of neighbours *k* for the local models is different for different time delays, and for the highest correlation coefficient ($r$=0.63) it is between $k$=75 and 100 neighbours corresponding to time delays between τ=8 and τ=12 hours. The experiments further showed that using a smaller number of neighbours ($k$=10–25) gives a better estimation of the peaks of the rainfall, but leads to the overestimation of the smaller rainfall depths. A mixture of models using different number of neighbours showed the best predictive performance.

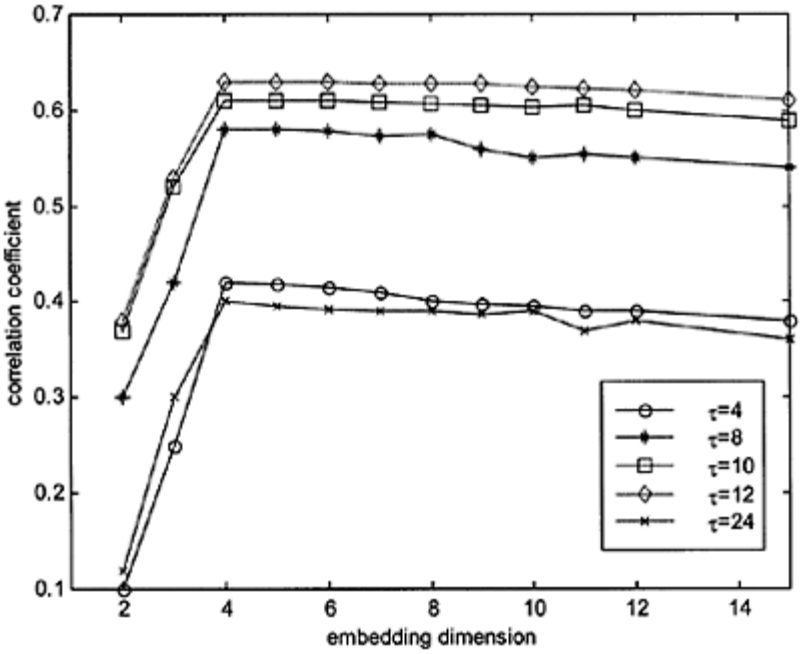



Figure 6.3.39. Correlation coefficient versus embedding dimension for different time delays. The prediction horizon is 1 hour ahead. Mixture of univariate local 3rd order polynomial models are used based on the reconstructed phase-space from the hourly rainfall time series.

The plots in Figure 6.3.39 show that the correlation coefficient increases to a maximum value when the embedding dimension is increased to $m$=4, which suggests that the correlation dimension of the attractor, and thus the embedding dimension, were correctly estimated. For noise-free chaotic dynamics the value of the maximum correlation coefficient between the measured and predicted rainfall should, in theory, remain unchanged (Casdagli, 1989). However, the plots in Figure 6.3.39 show that the prediction accuracy decreases slightly at higher embedding dimensions. Due to the presence of noise, the nearby points in the high-dimensional phase-space may be contaminated with points whose earlier coordinates (at low embedding dimensions) are close but whose recent coordinates (at high embedding dimensions) are distant. In other words they are false nearest neighbours, as already demonstrated in Figure 6.3.13 for the 15min data.

The prediction accuracy for the hourly rainfall data was checked by making predictions for different prediction horizons. Figure 6.3.40 shows the relationship between the correlation coefficient between the predicted and observed hourly rainfall

and the prediction horizon. It can be seen that the correlation coefficient decreases sharply with an increase in the prediction horizon. Such a decrease of the prediction accuracy for several hours in the future is due to the presence of chaotic dynamics expressed with a large positive Lyapunov exponent; see Figure 6.3.26. The prediction accuracy of the hourly rainfall for 1 hour ahead expressed with the correlation coefficient is $r$=0.63 using a mixture of univariate local models, which is rather good, taking into account the difficulty and complexity of the rainfall prediction. This prediction accuracy may be increased by employing multivariate local models, involving other variables, such as sun radiation, wind, temperature, humidity, air pressure and others. Figure 6.3.41 shows the observed and predicted rainfall data with a scatter plot for a part of the testing data set.

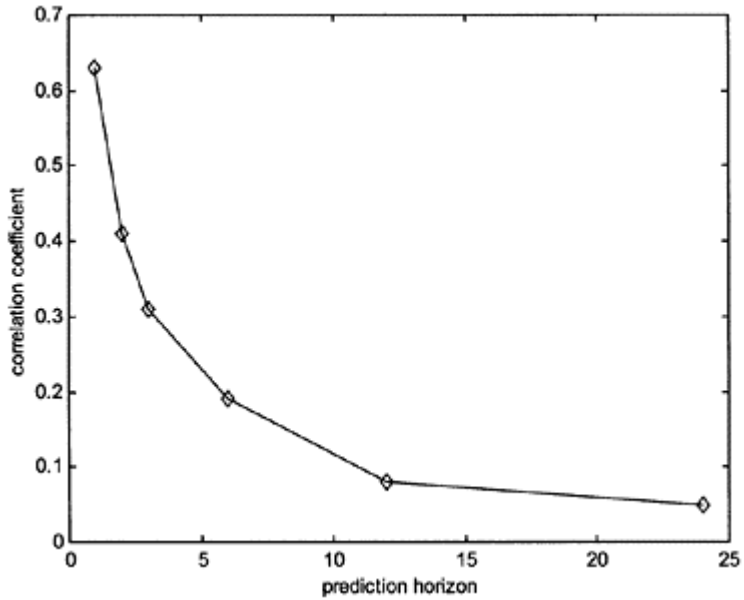



Figure 6.3.40. Correlation coefficient versus the prediction horizon. Mixture of univariate local 3rd order polynomial models are used based on the reconstructed phase-space from the hourly rainfall time series ($m$=4 and $\tau$=12).

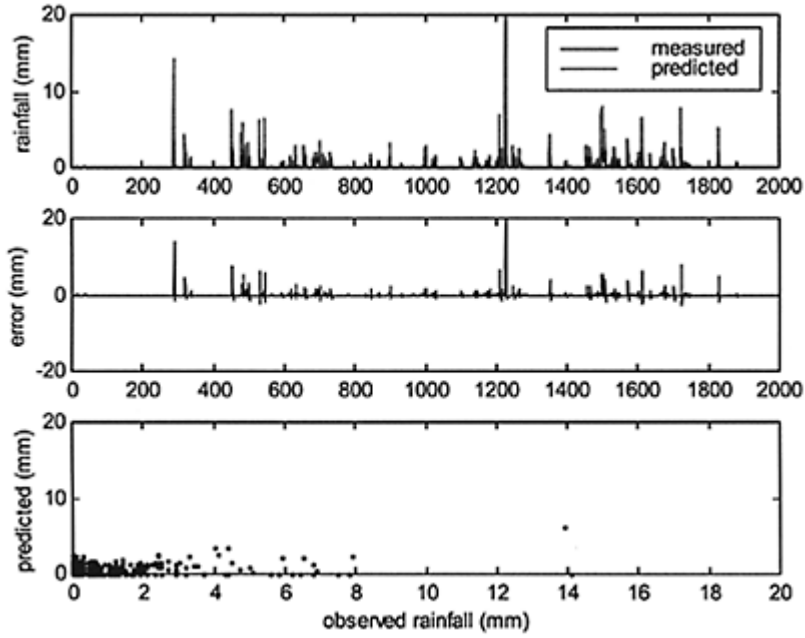



Figure 6.3.41. Observed and predicted hourly rainfall data. Prediction horizon is 1 hour with a correlation coefficient of $r$=0.63. A part of the testing data set is visualised where an extreme rainfall of 20 (mm) occurred.

A similar analysis was carried out to assess the predictive accuracy for the daily rainfall dynamics. Figure 6.3.42 shows the relationship between the correlation coefficient between the predicted and observed daily rainfall and the prediction horizon. Figure 6.3.43 shows the observed and predicted daily rainfall data with the error and the scatter plot for a part of the testing data set.

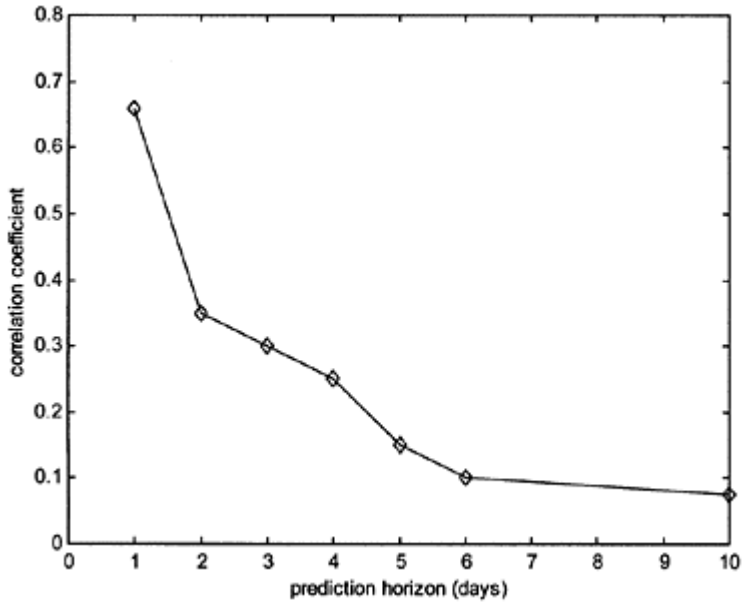



Figure 6.3.42. Correlation coefficient versus the prediction horizon. Mixture of univariate local 3rd order polynomial models are used based on the reconstructed phase-space from the daily rainfall time series ($m$=1 1 and $\tau$=4).

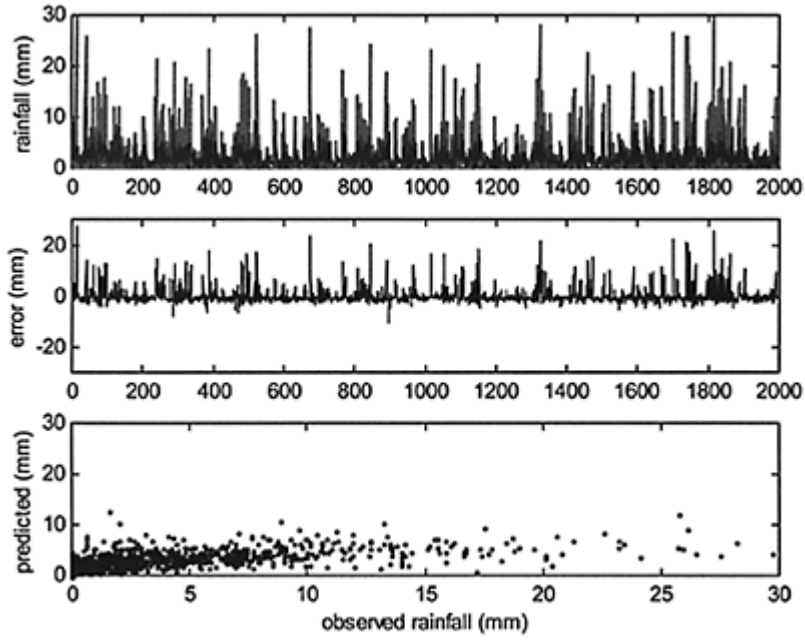



Figure 6.3.43. Observed and predicted daily rainfall data. Prediction horizon is 1 day with a correlation coefficient of $r$=0.66. A part of the testing data set (2000 days) is visualised.

Figure 6.3.42 shows that the correlation coefficient decreases with an increase in the prediction horizon. There is a substantial loss of prediction accuracy after the first day in the future due to the presence of hyper-chaotic dynamics. The prediction accuracy for the daily rainfall for one day ahead expressed with the correlation coefficient is $r$=0.66.

### *6.3.5 Discussion and conclusions*

In this application we have investigated the existence of chaos in rainfall dynamics using the methods and techniques from nonlinear dynamics and chaos mathematics, based on the rainfall time series recorded at the De Bilt meteo station in the Netherlands. The main question of the existence of structurally different chaotic dynamics in the rainfall using different temporal scales of the observables was addressed by the analysis of 15min, hourly, daily and weekly rainfall data.

The correlation dimension method provided evidence of the existence of a low-dimensional attractor for the different rainfall data sets aggregated over different time periods, thus suggesting the existence of chaotic dynamics. Based on the attractor dimensions that resulted for the 15min, hourly, daily and weekly rainfall data, the

minimum number of variables essential to model the rainfall dynamics was identified as 3,4,11 and 11, respectively. The indicative number of sufficient variables to fully describe the rainfall dynamics on different temporal scales was identified as 40, 38, 30 and 11, respectively. The effects of the time delay value, used for the phase-space reconstruction, on the attractor dimension estimation were also investigated in order to compare the results obtained from the average mutual information function.

The Lyapunov exponents computed on the 15min, hourly, daily and weekly rainfall data, demonstrated strong evidence of the existence of chaotic dynamics in the 15min and hourly data and hyper-chaos in the daily and weekly rainfall dynamics. The existence of positive Lyapunov exponents for all rainfall data sets clearly showed the limits of the predictability of any model.

The method of surrogate data for distinguishing between chaotic and stochastic rainfall dynamics based on the continuous wavelet transform, together with the test for nonlinearity, provided evidence that the rainfall dynamics is different from a linear stochastic process. In addition, the simple nonlinear noise-reduction algorithm applied to the different rainfall data sets improved the results for the correlation dimension estimation, and thus the reconstructed phase-space.

The nonlinear prediction method based on univariate local modelling in the reconstructed phase-space enabled us to check the prediction accuracy using different time horizons and with respect to: (i) number of neighbours; (ii) optimal time delay; and (iii) the embedded dimension. The results indicated a reasonable short-term predictability for the hourly and daily rainfall, but with a sharp drop in the prediction accuracy due to the presence of hyper-chaotic dynamics. The mixture of models framework, elaborated in Chapter 5, using different capacity for the models (experts), showed the best predictive performances.

In summary, the results from this application lead to the conclusion of the existence of structurally different chaotic dynamics in the rainfall at different temporal scales. However, rainfall is a multidimensional spatio-temporal phenomenon. The rainfall dynamics are not only highly fluctuating in time but also in space. These spatio-temporal signatures (patterns) are not independent but rather dependent. In addition, they usually occur at rather small grid resolutions, such as 5–10 km. Much more needs to be done in the collection of fine-resolution data in space and time in order to be able to study the spatiotemporal rainfall dynamics and to improve the numerical weather models. The recent advances in remote sensing and radar surveillance technology will help in the collection of such kind of data. At present it is not clear whether the dynamics of spatio-temporal rainfall patterns can be described by an attractor in a phase-space over a certain area.

## 6.4 Rainfall-runoff modelling

### *6.4.1 Introduction*

There is a continuing interest in hydrology to model the relationships between rainfall and runoff. The issue of developing faster and more accurate rainfall-runoff models still occupies one of the central areas in the research-orientated hydrological community. At

the same time, the application-orientated part of the community requires even simpler, transparent, but still acceptably accurate models, especially for the purposes of flood forecasting and management.

Different types of rainfall-runoff transformation models have been proposed, ranging from purely empirical simple models, such as the rational method, to highly sophisticated distributed physical process models defined by partial differential equations, such as SHE (System Hydrologique Europeèn) model (Abbott et al., 1986). Based on the conceptualisation and the degree of representation of the involved physical processes, the models are classified, with the increasing degree of representation, as black-box models, conceptual models, and physically-based distributed models. A detailed overview of the characteristics of these three classes of hydrological models and their modelling protocols can be found, for example, in Refsgaard and Knudsen (1996), Velickov (1998).

Black-box models are often used because they (partially) avoid having to address the problems of the spatial and temporal variability of the inputs and parameters, and the complexities of the involved physical processes. The unit hydrograph is one such linear black-box model and has been widely accepted in the past as a practical tool. However, the limitation of the unit hydrograph in representing the rainfall-runoff relationship is not only because of the restrictions of linearity and time invariance but is also because of the uncertainties in the determination of the "effective rainfall" and the separation of "baseflow" (Brath and Rosso, 1993). Hence, non-linearity was introduced as Volterra integral series for the analysis of hydrologic systems (for an overview, see Singh, 1988). In the early of 1990s, the quest for more accurate, but still relatively simple-to-use, black-box models has been reinforced by yet another technique, namely that of artificial neural networks. The encouraging results obtained by many hydrologists, applying ANNs on various different catchments (see e.g. Hsu et. al., Hall and Minns, 1993), have clearly indicated a number of desirable properties that ANNs offer, and they have become the *defacto* data-driven modelling tool in practice. Further studies (e.g. Solomatine and Torres, 1996; Minns, 1998, Velickov 1997, Dibike 2002) has proven that ANNs indeed represent a valuable rainfall-runoff modelling paradigm.

The theory of nonlinear dynamics and the concept of deterministic chaos has recently motivated applications in rainfall-runoff modelling, such as the reconstruction of the runoff dynamics and runoff forecasting. There is still an ongoing quest for the existence of deterministic chaos in the runoff dynamics. The initial investigation using the correlation dimension analysis on the Twin River daily runoff data (8458 points) by Savard (1990) showed the possible existence of an attractor with a correlation dimension of $d_c$=7.9. Jayawardena and Lai (1994) investigated further the daily streamflow data from two stations in Hong Kong. The geometric and dynamic invariants of the reconstructed runoff dynamics were applied to streamflow data sets consisting of 7300 and 6205 points respectively. Their study demonstrated initial evidence of the possible existence of a low-dimensional attractor in the runoff dynamics. Lai Porporato and Ridolfi (1996) provided evidence of the existence of deterministic chaos in the daily flow data of Dora Baltea, a tributary river of the river Po, in Italy. The initial application of the correlation dimension method to a time series consisting of 14,246 daily flows indicated the existence of low-dimensional attractor in the runoff dynamics. This study was further extended (Porporato and Ridolfi, 1997) with noise reduction, synthetic data application, and nonlinear prediction using univariate local models, which provided important

confirmations about the nonlinear behaviour of the flow phenomenon. Lui *et al.* (1998) analysed the daily streamflow data observed at 28 selected stations in United States and reported that the daily streamflow dynamics is characterised by a wide dynamical range between deterministic chaos and periodic signals contaminated with additive noise. Wang and Gan (1998) further analysed the unregulated streamflow data of six rivers in the Canadian prairies and found correlation dimensions of the attractors between $d_c$=4–7. Babovic and Keijzer (1999) applied correlation dimension and Lyapunov exponent methods to the daily runoff data (1826 data points) of the river Luznice in Czech Republic. The authors reported evidence of the existence of chaotic dynamics, but did not find a clear signature of the attractor, due to limited amount of data (insufficient statistics for higher embeddings) and the presence of noise in the data. However, univariate local modelling based on the runoff data showed a good performance in comparison with the genetic programming technique.

The main objective of this case study is to investigate further the character of the runoff dynamics using the elaborated nonlinear methods, and to contribute to the rainfall-runoff modelling using a novel multivariate local modelling approach. The first part of the case study focuses on an exploratory data analysis and reconstruction of the runoff dynamics based on the daily time series of runoff and rainfall for the period of 1976–1996, applying the techniques for multivariate phase-space reconstruction, elaborated in Section 3.3. This analysis is further extended with the identification of the existence of different hydrologic dynamical regimes (responses) in generating the runoff from the catchment. In the second part, we demonstrate the applicability of the multivariate local models in phase-space, together with the mixture of models framework, for modelling the rainfall-runoff dynamics. In addition, we compare the results with those of univariate local models and artificial neural network rainfall-runoff models, based on research conducted by Chuanbao (2001) and Shrestha (2002) in the framework of this thesis.

### *6.4.2 The study area and available data*

The catchment selected for this study is the upper reach of the Huai River basin, namely Xixian sub-catchment, located upstream of the Xixian station in East China; see Figure 6.4.1. The catchment area is about 10,190 ($km^2$). Most of the area is mountainous, especially the western and southern parts are more undulating with the highest peak reaching 1,140 (m) above the mean sea level. To large extent the catchment is covered with vegetation and forestation. There is also a reservoir in the catchment, which was built at the end of 1950's in the Shi River, a branch of the Huai River. Its controlling area is 1,100 ($km^2$) that takes account of about 10% of the total area of the catchment. Figure 6.4.1 shows the river networks and the distribution of rainfall stations and discharge stations within the study area of the catchment. Because the moisture of the air in the Huai River basin is generally transported from the Yellow Sea in the east and the topography downstream of the Xixian is a broad plain, the mountains in upstream provide orographic lifting. This makes the Xixian catchment a storm centre with the highest annual rainfall reaching 1,500 (mm). Thus, this area is the main flood source of the Huai River basin. Approximately once in every 5 years the basin is seriously threatened and damaged by heavy floods originating from the Xixian catchment. The precipitation events during the monsoon season (May to September) can be characterised as moving

depressions and cyclones according to their causality. The latter type usually has the characteristics of higher intensity, shorter duration and smaller scope compared with the former.

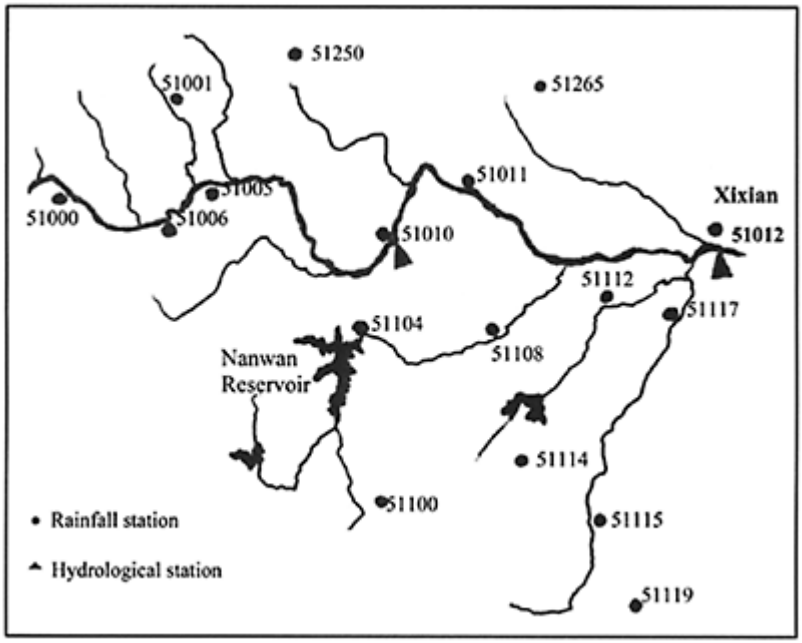



Figure 6.4.1. The map of the studied area

The mean annual rainfall over the basin is about 900 (mm) for the period of 1954–1996. It is normally larger in the southern parts, the hilly areas, and the coastal belt of the river basin. Figure 6.4.2 shows the distribution of the rainfall. The temporal variability of the rainfall is characterised by large seasonal change, which leads to droughts in the winter and spring and heavy rains in the summer and autumn. The seasonal average values of the spring, summer, autumn and winter for the period of 1954–1996 are estimated as 190 (mm), 490 (mm), 165 (mm) and 66 (mm) respectively.

The mean annual runoff is about 230 (mm/year), which corresponds to an average runoff coefficient of $r$=0.25. The actual runoff in the catchment varies in time and space. The spatial variation is between 50 (mm) and 1,000 (mm) from north to south (see Figure 6.4.3). The ratio between the maximum and minimum annual runoff varies between 5 and 30, with the higher value in the north of the basin. The yearly distribution of runoff is characterised by a concentration of runoff in the flood season (50%~88% of the annual value).

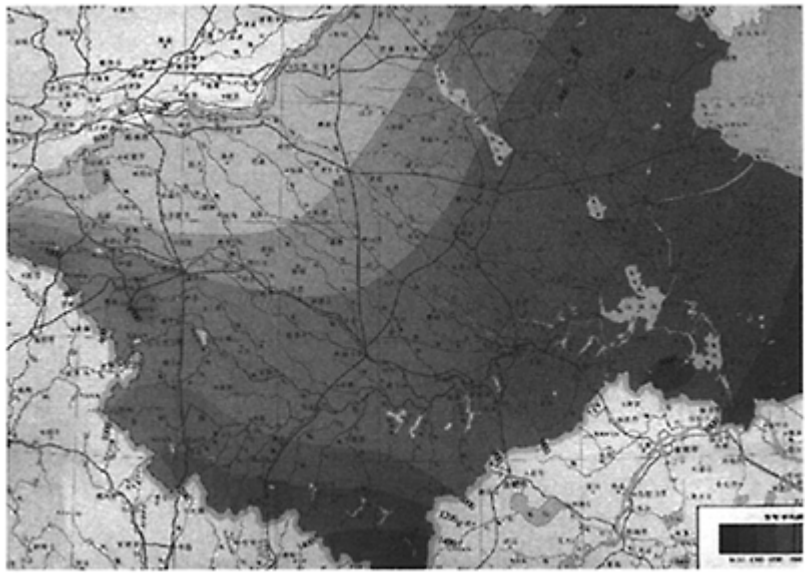


Figure 6.4.2. The isohyetal map of the mean annual precipitation.

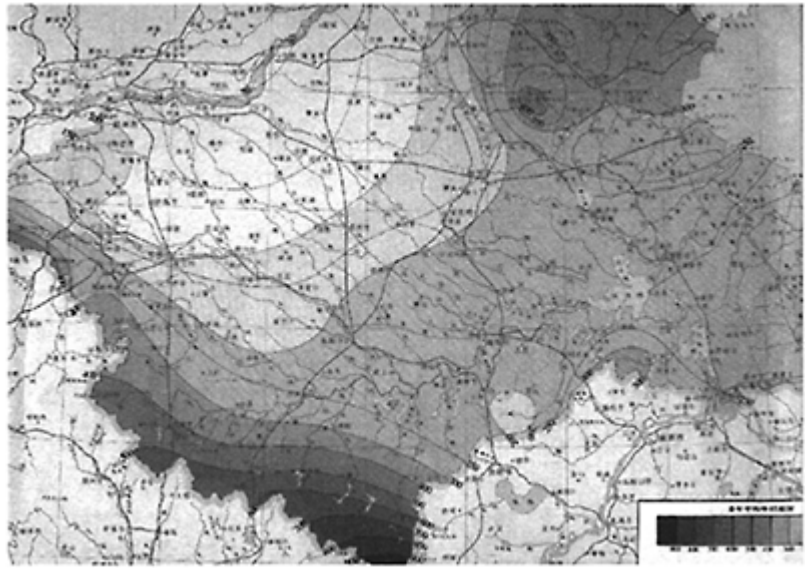


Figure 6.4.3. The spatial distribution of the annual runoff.

In this case study, the daily precipitation data from 17 rainfall stations and the daily discharge from 3 discharge stations are available for the period of 21 years from 1976 to 1996, resulting in time series with a length of $N$=7671 samples each. The Ministry of Water Resources in P.R.China provided the data for research purposes. In this period, 8 years are relatively wet, 6 years are relatively dry, and the remaining years are intermediate. The average annual rainfall and runoff from the catchment at Xixian station are presented in Table 6.4.1 and Figure 6.4.4.

Table. 6.4.1. Annual rainfall and runoff from 1976 to 1996

| Year | Rainfall (mm) | Runoff (mm) | Runoff coefficient | Year | Rainfall (mm) | Runoff (mm) | Runoff coefficient |
|---|---|---|---|---|---|---|---|
| 1976 | 745 | 174 | 0.23 | 1987 | 1498 | 714 | 0.48 |
| 1977 | 1207 | 378 | 0.31 | 1988 | 816 | 197 | 0.24 |
| 1978 | 787 | 125 | 0.16 | 1989 | 1248 | 477 | 0.38 |
| 1979 | 1169 | 328 | 0.28 | 1990 | 986 | 301 | 0.30 |
| 1980 | 1234 | 510 | 0.41 | 1991 | 1278 | 612 | 0.48 |
| 1981 | 974 | 257 | 0.26 | 1992 | 812 | 150 | 0.19 |
| 1982 | 1301 | 653 | 0.50 | 1993 | 989 | 228 | 0.23 |
| 1983 | 1133 | 450 | 0.40 | 1994 | 929 | 182 | 0.20 |
| 1984 | 1214 | 476 | 0.39 | 1995 | 918 | 216 | 0.24 |
| 1985 | 871 | 291 | 0.33 | 1996 | 1220 | 523 | 0.43 |
| 1986 | 833 | 156 | 0.19 | average | 1055 | 352 | 0.33 |

In order to calculate the average daily rainfall time series for the whole catchment from the 17 rainfall data, the Thiessen polygon method was used, assigning different weights to each rainfall station based on their coverage and geographic characteristics. The weighted sum of the 17 daily rainfall time series gives the average daily rainfall of the catchment using the following relation:

$$\overline{R} = \sum_{i=1}^{17} W_i R_i \tag{6.4}$$

where, $\overline{R}$ is the average daily rainfall, $W_i$ is the weight of the rainfall station $i$ and the $R_i$ is the daily rainfall of the rainfall station $i$.

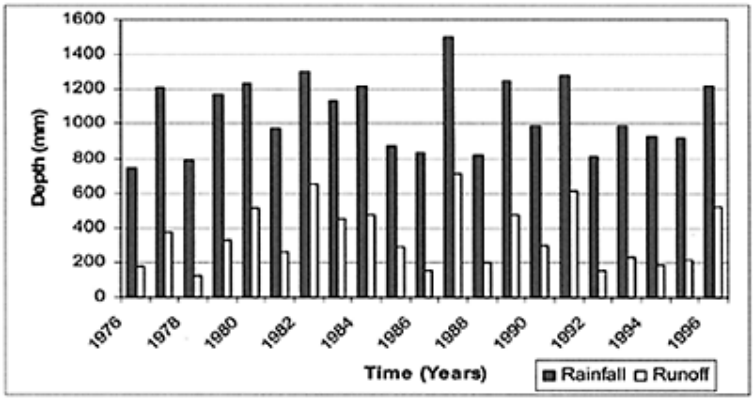



Figure 6.4.4. Annual rainfall and runoff from 1976 to 1996

*6.4.3 Exploratory data analysis*

The available data was first screened to check for the obvious errors. Two major checks conducted focused on: (i) recorded hydrographs whose volume exceeds the recorded rainfall, or is such a high fraction of the recorded rainfall as to be highly improbable (a condition usually associated with major storms over the watershed but largely missing the precipitation gauges), and (ii) large recorded rainfalls simultaneous with little runoff, a condition usually associated with intense rainfalls registered at the gauge not being representative of precipitation over the watershed. Figure 6.4.5 shows one of such plots.

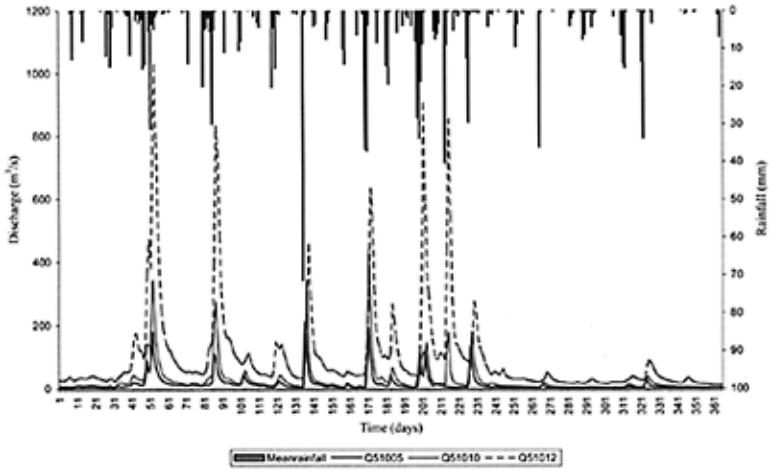



Figure 6.4.5. Rainfall and discharge for the studied area in 1990

In order to check the stationarity of the available data, the time series of daily flow from the three hydrological stations and the daily rainfall at all 17 rainfall stations were divided into three sub-sets, on which the basic statistical parameters were calculated. The following Table 6.4.2 presents the basic statistics for the three data sub-sets. The daily flow data of the three discharge stations and the 17 rainfall stations were further checked for consistency and homogeneity (Chaunbao, 2001) using double mass and correlation analysis. The result showed that data are consistent and the rainfall data originate from a climatic homogenous region.

The spatial distribution of rainfall over the studied area was analysed by the division of the catchment into 5 hydrological response units (HRUs), based on the watershed delineation and its geographic and orographic characteristics. The average daily rainfall for each HRU is calculated by the Thiessen method from the rainfall stations belonging to the unit under consideration. The basic statistics of the daily rainfall for the HRUs are shown in Table 6.4.3 and Figure 6.4.6.

Table 6.4.2. Basic Statistics of daily rainfall (in mm) and discharge data (in $m^3/s$). Pxxxxx and Qxxxxx denote the rainfall and runoff data of the station xxxxx, respectively.

| Variables | 1978–1982 | | | 1990–1996 | | | 1983–1989 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Length | Mean | Std. Dev. | Length | Mean | Std. Dev. | Length | Mean | Std. Dev. |
| P51000 | 2557 | 3.26 | 11.70 | 2557 | 3.36 | 14.53 | 2557 | 3.01 | 10.84 |
| P51001 | 2557 | 2.45 | 8.66 | 2557 | 2.30 | 8.81 | 2557 | 2.07 | 7.60 |
| P51250 | 2557 | 2.87 | 12.09 | 2557 | 2.59 | 9.67 | 2557 | 2.54 | 9.48 |
| P51005 | 2557 | 2.61 | 9.67 | 2557 | 2.55 | 10.35 | 2557 | 2.41 | 8.76 |
| P51006 | 2557 | 3.30 | 12.06 | 2557 | 3.40 | 12.00 | 2557 | 3.19 | 10.72 |
| P51010 | 2557 | 2.64 | 9.69 | 2557 | 2.63 | 9.82 | 2557 | 2.53 | 8.89 |
| P51265 | 2557 | 2.61 | 9.59 | 2557 | 2.53 | 9.42 | 2557 | 2.48 | 9.39 |
| P51104 | 2557 | 3.28 | 12.10 | 2557 | 3.04 | 10.08 | 2557 | 3.02 | 9.65 |
| P51011 | 2557 | 2.33 | 8.43 | 2557 | 2.57 | 9.56 | 2557 | 2.13 | 7.75 |
| P51012 | 2557 | 2.59 | 9.16 | 2557 | 2.70 | 9.30 | 2557 | 2.70 | 9.42 |
| P51112 | 2557 | 2.71 | 9.14 | 2557 | 2.81 | 9.71 | 2557 | 2.72 | 10.19 |
| P51108 | 2557 | 3.03 | 10.83 | 2557 | 3.00 | 10.21 | 2557 | 2.69 | 8.61 |
| P51100 | 2557 | 3.26 | 11.00 | 2557 | 3.69 | 13.59 | 2557 | 3.46 | 11.70 |
| P51114 | 2557 | 3.34 | 12.13 | 2557 | 3.52 | 12.49 | 2557 | 3.21 | 10.97 |
| P51117 | 2557 | 2.83 | 10.12 | 2557 | 2.87 | 10.03 | 2557 | 2.86 | 10.10 |
| P51115 | 2557 | 3.07 | 10.10 | 2557 | 3.51 | 13.06 | 2557 | 3.15 | 10.69 |
| P51119 | 2557 | 3.15 | 10.96 | 2557 | 3.49 | 12.93 | 2557 | 3.36 | 11.17 |
| Q51012 | 2557 | 19.62 | 70.09 | 2557 | 20.06 | 74.72 | 2557 | 14.29 | 47.15 |
| Q51010 | 2557 | 35.82 | 107.61 | 2557 | 36.78 | 124.41 | 2557 | 28.13 | 90.26 |
| Q51005 | 2557 | 111.84 | 294.23 | 2557 | 127.35 | 300.15 | 2557 | 102.03 | 267.18 |

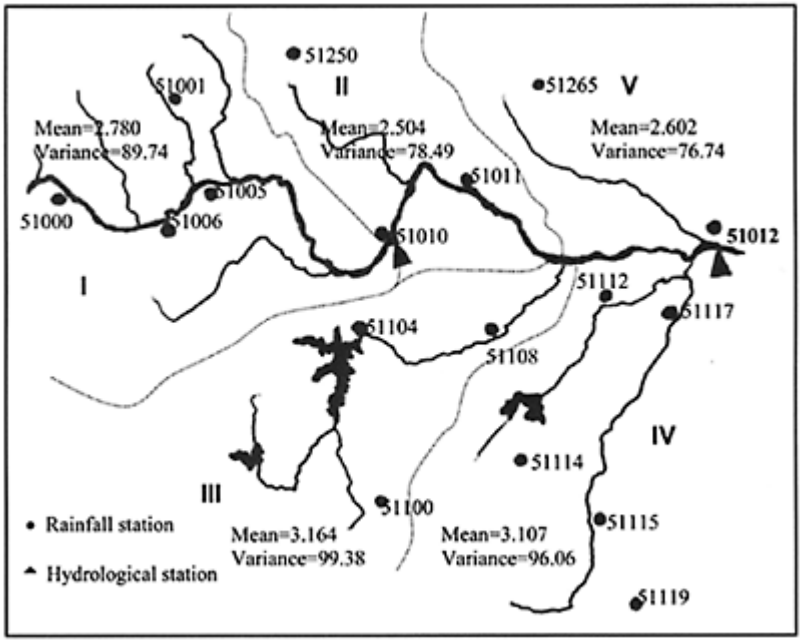



Figure 6.4.6. The map of the HRUs with the spatial distribution of the rainfall.

Table 6.4.3 and Figure 6.4.6 show that the HRUs III and IV, which include the stations from the southern part of the study area, have higher average rainfall as compared to the other units in the northern part. It shows the fact that the rainfall, driven by the monsoons, usually approaches from a south-eastrely direction and moves towards the north-west, and therefore the HRUs on the south will most likely be the first activated in the generation of the runoff from the catchment. Also there is a higher variability of rainfall in these parts as indicated by the higher values of the variance and the standard deviation.

Table 6.4.3. Basic Statistics of daily rainfall for 5 HRU in (mm).

| HRU | Stations (No) | Max | Min | Mean | Std. Dev. |
|---|---|---|---|---|---|
| I | 51000, 51001, 51005, 51006, 51010 | 229.72 | 0 | 2.78 | 9.47 |
| II | 51250, 51011 | 178.7 | 0 | 2.50 | 8.86 |
| III | 51014, 51108, 51000 | 156 | 0 | 3.16 | 9.97 |
| IV | 51112, 51117, 51114, 51115, 51119 | 173.04 | 0 | 3.10 | 9.8 |
| V | 51265, 51012 | 137.75 | 0 | 2.60 | 8.76 |

In order to quantify some of the temporal characteristics of the rainfall in the study area for the recorded period of time, the duration of the rainfall was analysed further. The frequency distribution of the daily rainfall duration is presented in Figure 6.4.7. The results demonstrate that 54.2% of the days in the analysed period (1976–1996) are dry, 1-day rainfall is 45.8%, 2-day rainfall is 32.0%, 3-day rainfall is 22.4%, 5-day rainfall is 11.3%, 7-day rainfall is 6.0% and more than 7-day rainfall is 4.5%. Based on the analysis of the selected flood hydrographs (see further Section 6.4.5), most of the floods in the catchment are caused by more than a 3-day rainfall, which, as reflected by the rainfall duration analysis, indicates a very high risk of flooding in the study area.

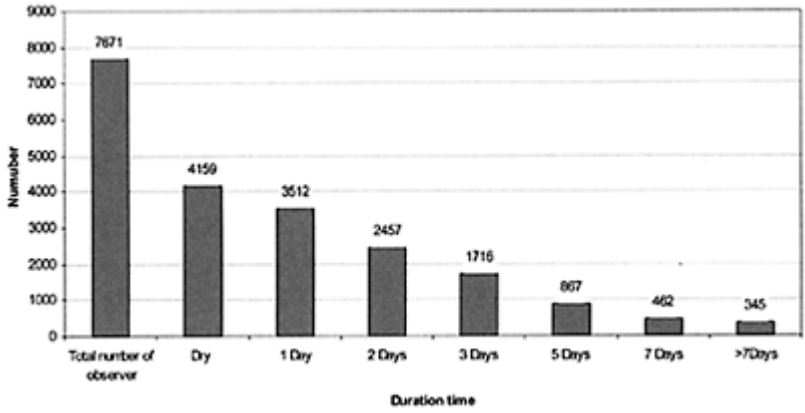



Figure 6.4.7. Distribution of the rainfall duration.

The spatio-temporal relationships between the mean daily rainfall and the daily discharges at the three stations were further investigated using cross-correlation analysis and the average mutual information technique. Figure 6.4.8 shows the correlation coefficient between the mean daily rainfall and the discharges at the upstream stations (51005, 51010) with the discharge at the target station (51012) for different time lags. Figure 6.4.9 shows the average mutual information between the same variables. Both, the cross-correlation and the AMI functions indicate time lag between the discharges of one day. The maximum cross-correlation coefficient between the mean daily rainfall and the discharge at the target station indicates a time lag of two days, whereas the AMI function indicates a time lag of 3–4 days, which demonstrates their nonlinear relationship.

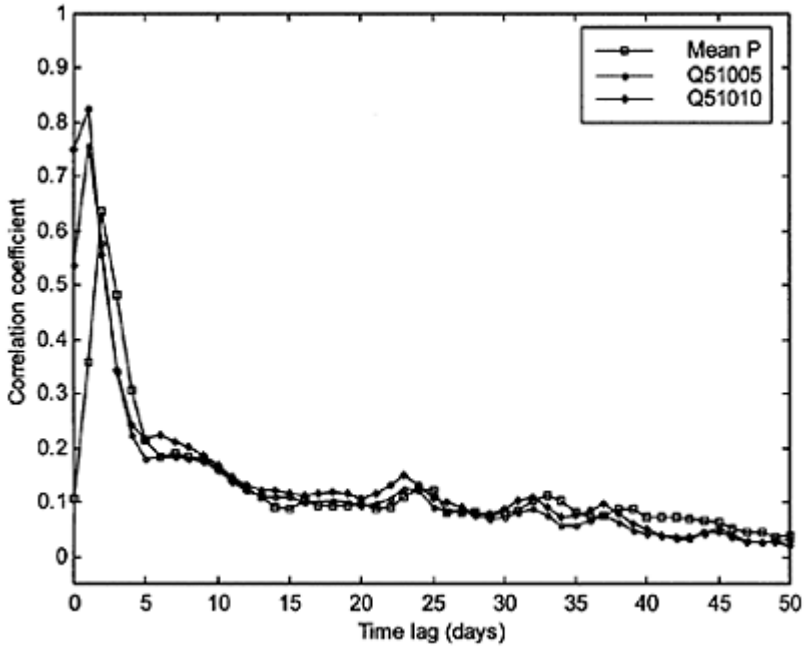



Figure 6.4.8. Cross-correlation function between the mean daily rainfall and the discharges at the upstream stations with the discharge at the target station (51012).

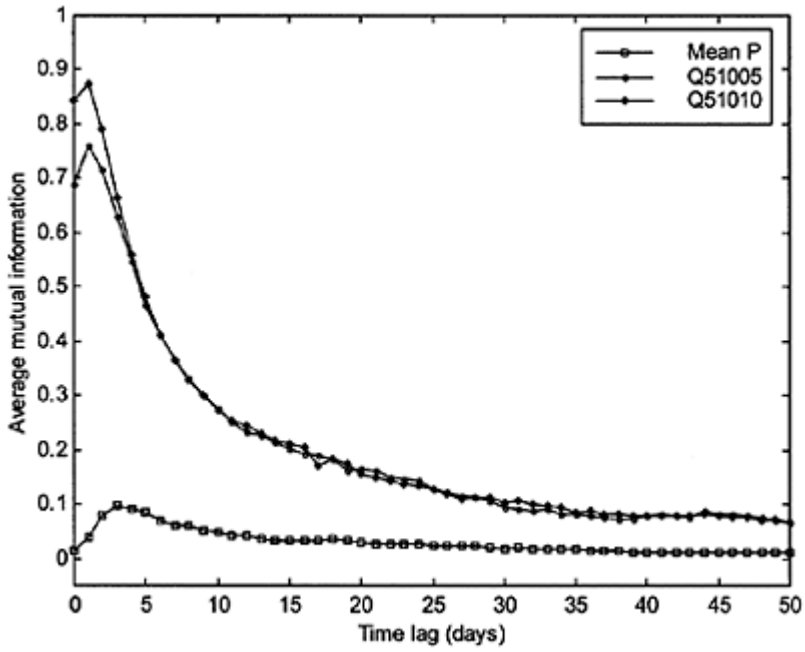



Figure 6.4.9. Average mutual information function between the mean daily rainfall and the discharges at the upstream stations with the discharge at the target station (51012).

The average mutual information function between the daily rainfall of each HRUs and the daily discharge at the target station 51012 is presented in Figure 6.4.10. The results indicate different time lags for the different HRUs, varying between 2, 3 and 4 days for the HRU4, HRU2 and HRU5, and HRU1 and HRU3, respectively.

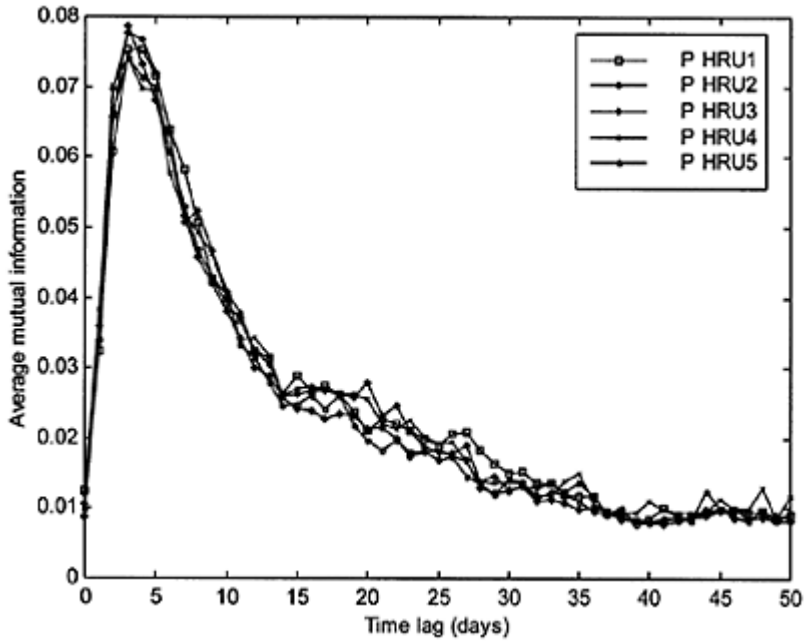



Figure 6.4.10. Average mutual information function between the spatially distributed daily rainfall at different HRUs and the discharge at the target station (51012).

### *6.4.4 Reconstruction of the dynamics from the discharge and rainfall observables*

The time series of the discharge and rainfall data were used to reconstruct the phase-space of the dynamical system using the methods and techniques elaborated in Chapter 3. The spectral analysis for both, the daily discharge and the mean daily rainfall (Figure 6.4.11) show broadband power spectra serving as first indicative signs of their chaotic dynamics.

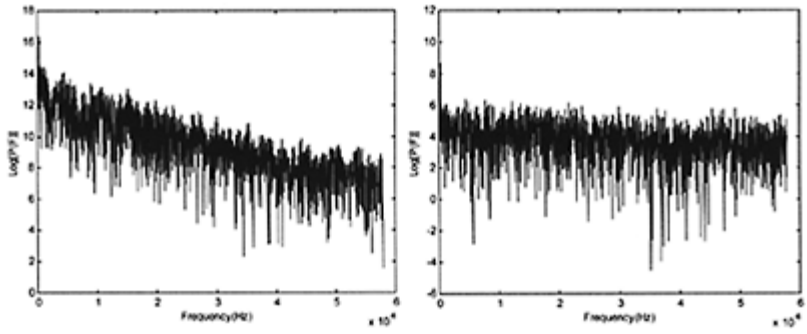



Figure 6.4.11. Power spectrum for the: (a) daily discharge at 51012 and (b) the mean daily rainfall.

The power spectrum of discharge station 51012 shows only one dominant periodicity, which is the annual periodicity of 372 days. The rest of the power spectrum is very broad indicating weak periodicities at 152, 108, 87, 26, 13 and 7 days. Similarly, the power spectrum of the mean daily rainfall indicates a yearly periodicity driven by the monsoons occurrence during the summers while the rest of the power spectrum is characterised with a broad noise-like spectrum with no dominant periodicities.

The autocorrelation and average mutual information functions for the daily flows at the three discharge stations and the daily mean rainfall are presented in Figure 6.4.12.

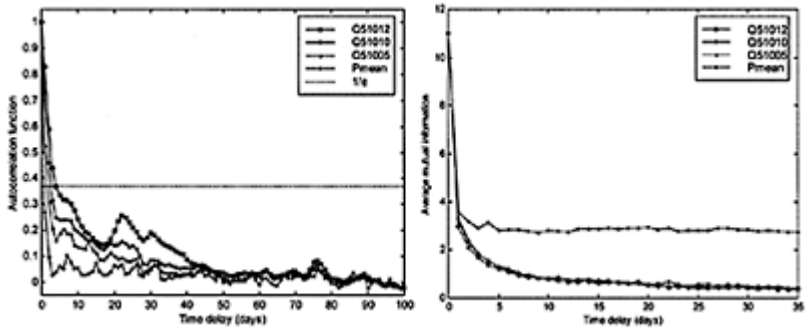



Figure 6.4.12. The autocorrelation (left figure) and average mutual information (right figure) functions for the daily discharges and the mean daily rainfall.

The autocorrelation function for the discharge at the target station 51012 shows an annual cycle (on an annual time scale) and small seasonal periodicities indicating the presence of a certain deterministic (but not dominant) runoff generation mechanism. For the exponentially decaying autocorrelation function for the discharge (long system memory),

the value of $1/e$ corresponds to time delay $\tau$=5 days. The first minimum of the average mutual information suggests a time delay for the reconstruction of the runoff dynamics of $\tau$=9 days. The autocorrelation function for the mean daily precipitation shows small periodicities on a weekly cycle, whereas the average mutual information indicates a substantial loss of information after the first day, reaching the first minimum at $\tau$=3 days.

The correlation dimension $d_c$ for the discharge at the target station 51012, which is used to assess the embedding dimension $m$, was estimated from the time series using the methodology described in Section 3.3.2. Figure 6.4.13 shows the correlation integral for the discharge data at station 51012 for different length scales. The relationship between the correlation exponent and the embedding dimension for different time delays is further presented in Figure 6.4.14. From Figure 6.4.14 there is a distinct saturation value of the correlation exponent for time delays between $\tau$=4 and $\tau$=12 days. The value of the correlation dimension of the attractor in this case is estimated to be $d_c$=3.20. Taking into account the discussion about the estimation of the embedding dimension $m$ (see Section 3.2.4), if we use Taken's embedding theorem the embedded dimension (integer number) of the manifold which contains the attractor is $m$=8. If we use Withney's recommendation, the embedding dimension is $m$=6. Abrabanel's recommendation (the first integer above the correlation dimension) leads to $m$=4. The false nearest neighbours method suggests an embedding dimension between $m$=5 and $m$=8; see Figure 6.4.15.

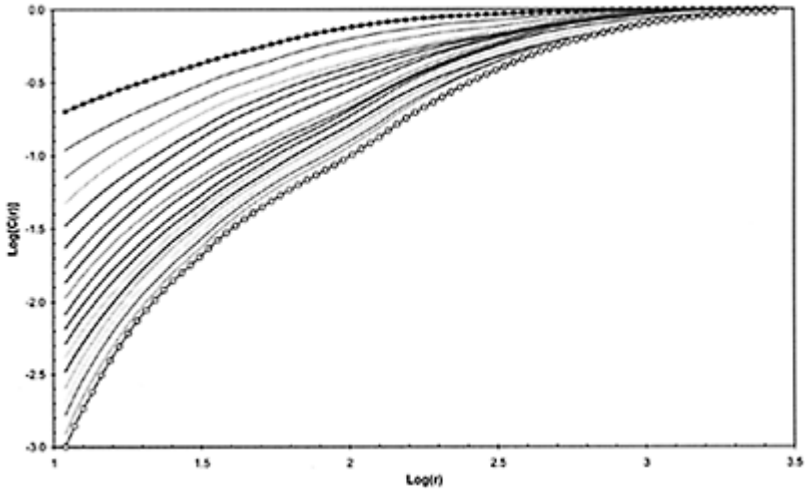



Figure 6.4.13. Correlation integral (sum) for the discharge data for the station 51012 (period 1971–1991, daily data). A double logarithmic plot was chosen for better visual presentation of the power law scaling between the correlation sum *C(r)* and the length scales *r*. The correlation sum was computed for different embedding dimensions (the line with squares corresponds to embedding dimension 2 and the line with open circles correspond to embedding dimension 20). The time delay used to produce this figure is $\tau$=8.

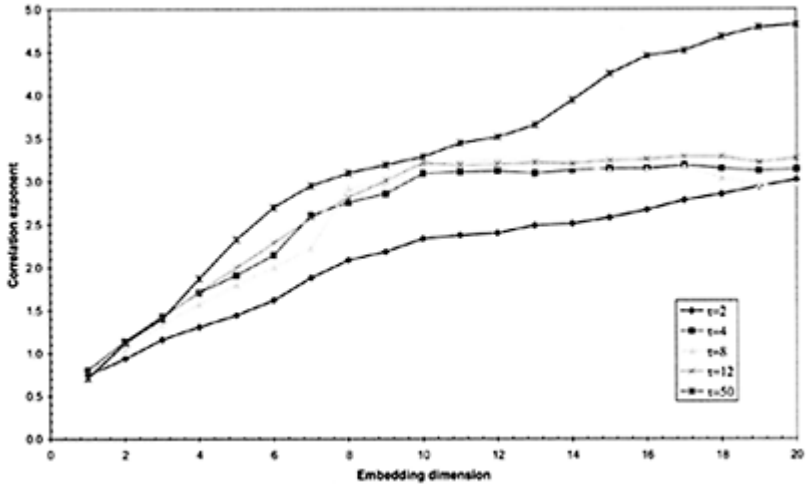



Figure 6.4.14. Relationship between the correlation exponent v and embedding dimension *m* for the discharge time series at the target station 51012 using different time delays τ. The correlation exponent increases with an increase of the embedded dimension up to a certain value and further saturates (when using time delays between τ=4 and τ=12 days). The saturation value of the correlation exponent, that is the correlation dimension, is 3.2 (uncertainty 0.1), which indicates presence of an attractor in the runoff dynamics.

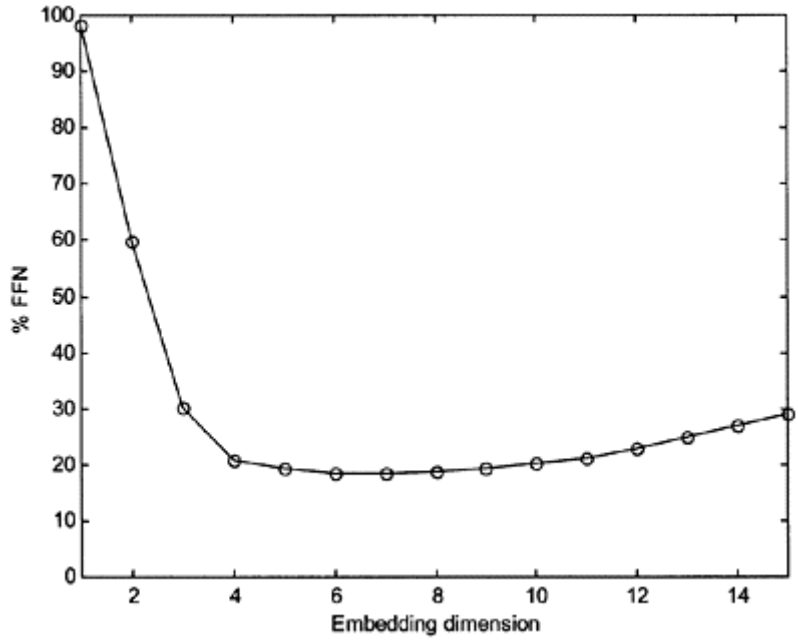



Figure 6.4.15. The percentage of the false nearest neighbours as a function of the embedding dimension for the discharge data at the target station 51012.

Similarly, the correlation dimension $d_c$ for the mean daily rainfall was assessed using the correlation sum analysis and is presented in Figure 6.4.16.

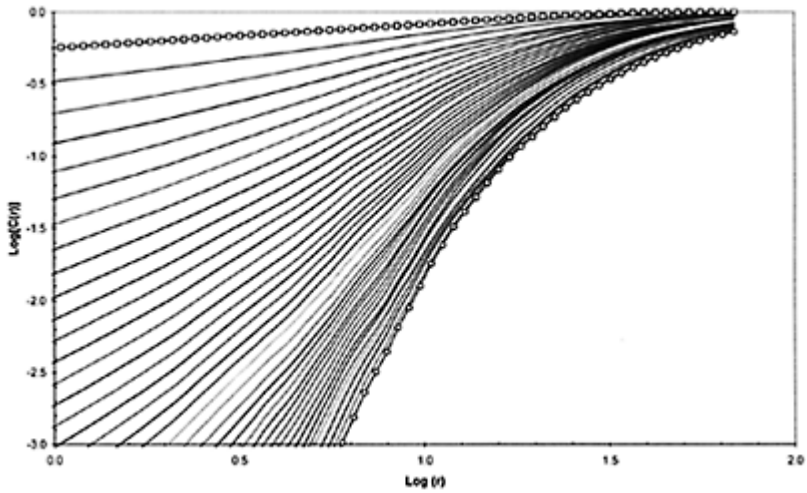



Figure 6.4.16. Correlation sum of the daily rainfall time series for different embedding dimensions, between $m$=2 (squares) and $m$=40 (circles). The time delay used to produce this figure is $\tau$=4 days.

The relationship between the average correlation exponent (correlation dimension) and the different time delays employed for the reconstruction of the phase-space based on the daily rainfall data is presented in Figure 6.4.17.

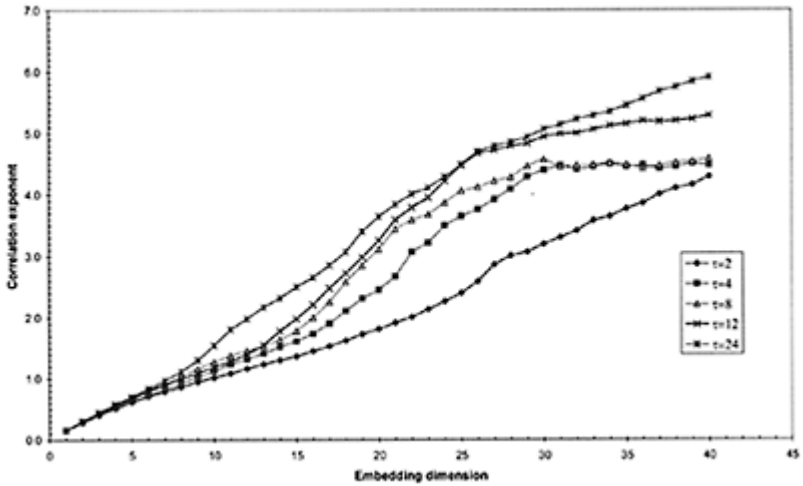



Figure 6.4.17. Relationship between the correlation exponent and the embedding dimension *m* for the daily rainfall time series using different time delays τ.

Figure 6.4.17 shows that the correlation exponent increases with an increase in the embedding dimension up to a certain value and further saturates (when using time delays between τ=4 and τ=8 days). The saturation value of the correlation dimension for the optimal time delay of τ=4 days, is $d_c$=4.6 (uncertainty 0.15) which indicates presence of an attractor in the daily rainfall dynamics. Application of Taken's embedding theorem suggests a dimension of the reconstructed phase-space (integer number) as $m=2d_c+1=10$, wheras Withney's recommendation suggests an embedding dimension of $m=9$ for the daily runoff dynamics. The saturation value of the correlation dimension occurs at an embedding dimension about $m$=31. These results are consistent with the results from the analysis of the daily rainfall time series for the De Bilt station. The false nearest neighbours method suggests an embedding dimension between $m$=8 and $m$=10; see Figure 6.4.18.

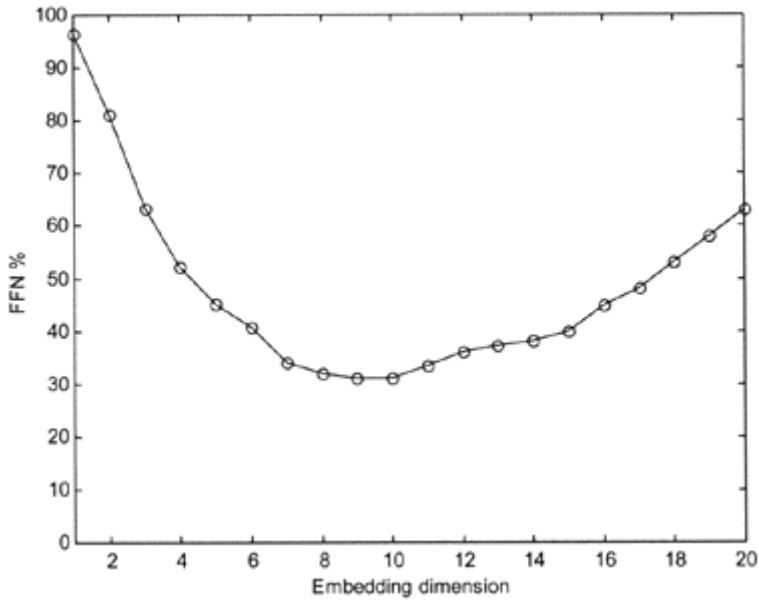



Figure 6.4.18. The percentage of the false nearest neighbours as a function of the embedding dimension for the mean daily rainfall.

The Lyapunov exponents, estimated for both the discharge and the rainfall daily time series, are presented in Figure 6.4.19. The largest Lyapunov exponent for the discharge at the target station 51012 is estimated as $\lambda_1$=1.21 (uncertainty 0.05), which indicates a loss of predictive information of 1.21 bits/day during the dynamical evolution of the system that yields a runoff predictability (based on the time series) of approximately one day. Whereas, for the mean daily rainfall times series, the presence of a higher value of the largest Lyapunov exponent ($\lambda_1$=5.3) indicates that the prediction horizon for the rainfall dynamics is very short.

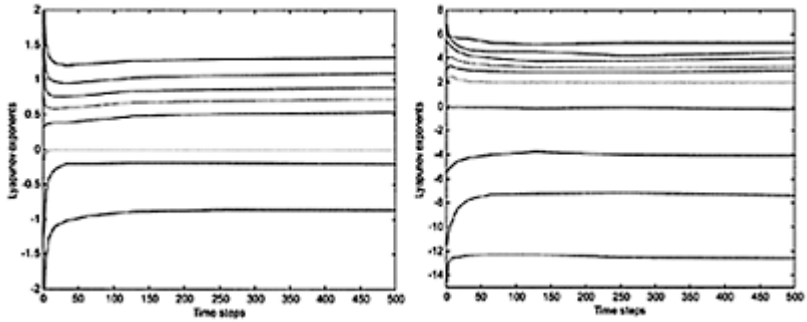


Figure 6.4.19. Lyapunov exponents for the daily discharge at the target station 51012 (left figure) and the mean daily rainfall (right figure).

Figure 6.4.19 further shows that the Lyapunov spectrum for the mean daily rainfall contains a large negative exponent $\lambda_{10}$=−12.40, which indicates presence of strong dissipation mechanisms in the rainfall dynamics. The presence of positive Lyapunov exponents and the fact that $\sum_{n=1}^{10} \lambda_n = -1.07 < 0$, provide strong evidence that the rainfall dynamics is driven by deterministic chaos. On the other hand, the Lyapunov spectrum estimated from the daily discharge time series at the target station 51012 shows a weaker dissipation mechanism. In addition, the sum of the Lyapunov exponents $(\sum_{n=1}^{8} \lambda_n = 3.36 > 0)$ is greater than zero, which indicates that the average rate of divergence of the small perturbations in the runoff dynamics is dominating the average rate of their convergence. In other words, the trajectory of the runoff dynamics in the reconstructed phase-space is not bounded indicating that the system may not be asymptotically stable. In physical terms this implies that the studied catchment is capable of generating excessive runoffs (not yet historically observed) and could also exhibit more asymptotically stable conditions (regimes). One possible explanation of this kind of deterministic hyper-chaotic behaviour is the highly nonlinear coupling of the processes underlying the runoff dynamics.

### *6.4.5 Investigation of existence of different dynamic regimes in the runoff*

The reconstruction of the runoff dynamics based on the times series of observables indicated the possible existence of different runoff generation mechanisms (dynamical regimes). In order to investigate the possible existence of different runoff generation mechanisms from the catchment, a classification analysis of the flood hydrographs with a peak discharge greater than Q>500 ($m^3/s$) (Chuanbao, 2001) was carried out. A

nonsupervised Bayesian classification technique, known as Autoclass developed by Cheeseman and Stultz (1994) and elaborated in a data mining context by Velickov and Solomatine (2000), was used to classify 89 hydrographs in total using the following features (see Figure 6.4.20):

$T_R$: Hydrograph rise time (day)
$T_L$: Hydrograph detention time (day)
$T_L/T_R$: Ratio of $T_L$ and $T_R$
$T_B$: Hydrograph base time (day)
$T_C$: The concentration time (day)
K: Runoff coefficient
$Q_{max}$: Peak discharges ($m^3/s$)
P: Total average rainfall (mm)
$P_i$: Rainfall intensity (mm/day)
W: Total runoff volume ($m^3$)

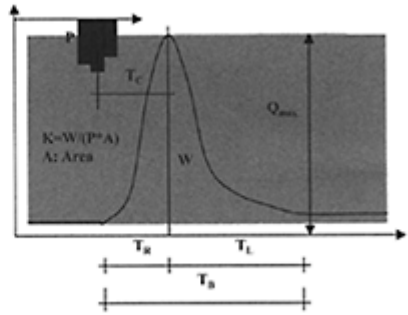



Figure 6.4.20. Features extracted from the hydrogpraphs used in the classification analysis.

On the basis of these features, the four classes of the runoff generation expressed through the hydrograph formation were identified as shown in Figures 6.4.21a–d. Class 1 (Figure 6.4.21a) is dominated by the hydrograph rise time, rainfall intensity, base time and detention time. This class has the same rise and detention time. It's occurrence is usually caused by the 2–3 days rainfall, and after the peak discharge there is no rainfall occurrence.

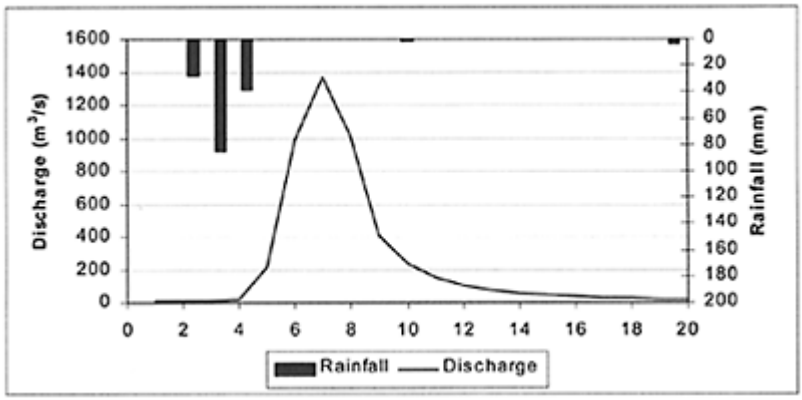



Figure 6.4.21a. A typical hydrograph for class 1. The figure shows flood event observed in 1978.

Class 2 is dominated by the rise time, base time, detention time and the ratio of detention time to rise time. This class has a relatively longer rise time, which is caused by the relatively longer duration of rainfall as shown in Figure 6.4.21b.

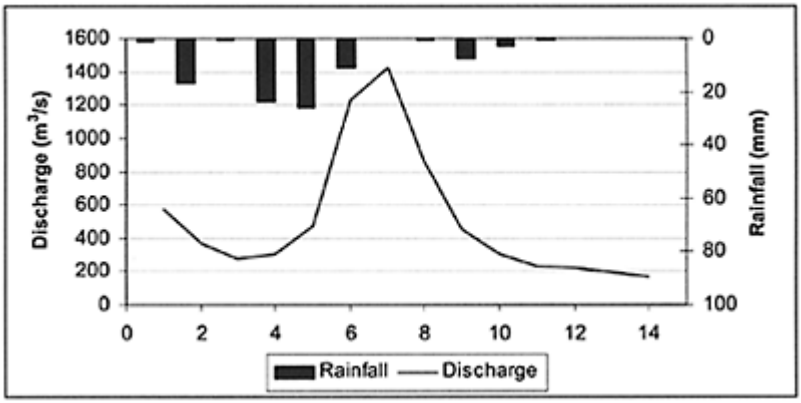



Figure 6.4.21b. A typical hydrograph for class 2. The figure shows flood event observed in 1977.

Figure 6.4.21c further shows a typical hydrograph of class 3, which has dominant features such as rise time, peak discharge, total average rainfall and ratio of detention time to rise time. This class has relatively longer tail, which is usually caused by intensive 2–3 days rainfall, and after the peak discharge there is usually less-intensive rainfall, which lasts for long period of time (10–12 days).

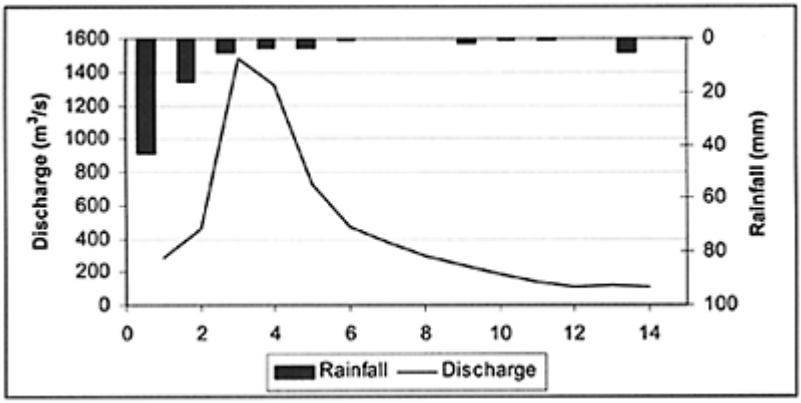



Figure 6.4.21c. A typical hydrograph for class 3. The figure shows flood event observed in 1991.

Lastly, class 4 is dominated by the rainfall intensity and duration, the rise time and the peak discharge. This class has sharp shape as shown in the Figure 6.4.21d, which is usually caused by very intensive rainfall. Although this kind of rainfall has duration of 2–3 days, it can generate an extreme flood, especially due to preparation (wetting) of the catchment by a previous rainfall.

The basic statistics of each class are shown in the Table 6.4.4, which demonstrates the differences in the features between the four identified classes. Thus, from the hydrograph analysis we can initially conclude the existence of possibly four different dynamic regimes in the generation of the flood hydrographs. This information is further used for the application of the mixture of models framework, which is discussed in the following sections.

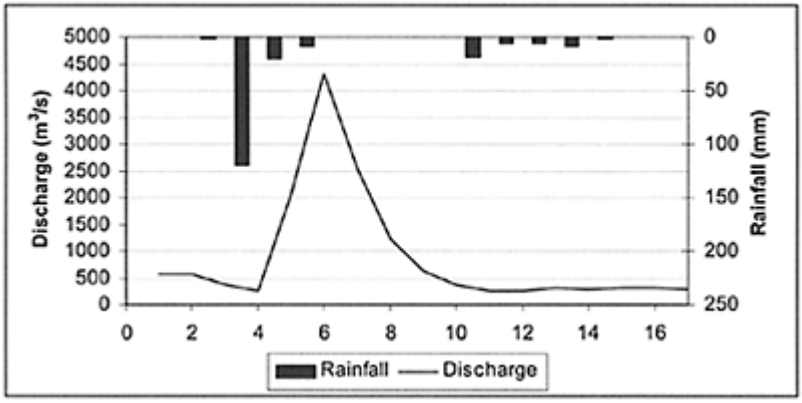



Figure 6.4.21d. A typical hydrograph for class 4. The figure shows flood event observed in 1980.

Table. 6.4.4. Basic statistics of each hydrograph class.

| Class | Feature | Num. of instances | Mean | Min | Max | Std. Dev. |
|---|---|---|---|---|---|---|
| 1 | $T_R$ | | 3.16 | 2.00 | 6.00 | 0.93 |
| | $T_L$ | | 5.63 | 3.00 | 10.00 | 1.57 |
| | $T_L/T_R$ | | 1.87 | 1.00 | 5.00 | 0.72 |
| | $T_B$ | | 8.79 | 5.00 | 14.00 | 2.08 |
| | $T_C$ | 31 | 2.50 | 0.00 | 4.00 | 0.71 |
| | K | | 0.48 | 0.19 | 0.95 | 0.19 |
| | $Q_{max}$ | | 1601 | 522 | 4070 | 1019 |
| | P | | 127.52 | 37.24 | 291.90 | 63.97 |
| | Pi | | 25.11 | 90.31 | 50.93 | 12.08 |
| 2 | $T_R$ | | 2.00 | 2.00 | 2.00 | 0.00 |
| | $T_L$ | | 4.00 | 4.00 | 4.00 | 0.00 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | $T_L/T_R$ | | 2.00 | 2.00 | 2.00 | 0.00 |
| | $T_B$ | | 6.00 | 6.00 | 6.00 | 0.00 |
| | $T_C$ | 27 | 2.06 | 1.50 | 3.00 | 0.29 |
| | K | | 0.40 | 0.16 | 0.82 | 0.16 |
| | $Q_{max}$ | | 1552 | 758 | 3620 | 769 |
| | P | | 118.52 | 47.41 | 215.94 | 38.72 |
| | Pi | | 33.91 | 13.20 | 86.57 | 19.06 |
| 3 | $T_R$ | | 2.00 | 2.00 | 2.00 | 0.00 |
| | $T_L$ | | 5.58 | 5.00 | 7.00 | 0.69 |
| | $T_L/T_R$ | | 2.79 | 2.50 | 3.50 | 0.35 |
| | $T_B$ | | 7.58 | 7.00 | 9.00 | 0.69 |
| | $T_C$ | 19 | 2.16 | 1.50 | 3.00 | 0.37 |
| | K | | 0.39 | 0.23 | 0.65 | 0.14 |
| | $Q_{max}$ | | 789 | 507 | 1480 | 283 |
| | P | | 82.10 | 59.29 | 112.34 | 16.06 |
| | Pi | | 31.86 | 11.85 | 88.99 | 19.46 |
| 4 | TR | | 1.67 | 1.00 | 2.00 | 0.49 |
| | $T_L$ | | 4.00 | 3.00 | 5.00 | 0.85 |
| | $T_L/T_R$ | | 2.58 | 1.50 | 4.00 | 0.82 |
| | $T_B$ | | 5.67 | 4.00 | 7.00 | 1.15 |
| | $T_C$ | 12 | 1.67 | 1.00 | 2.00 | 0.44 |
| | K | | 0.45 | 0.12 | 0.79 | 0.21 |
| | $Q_{max}$ | | 2622 | 618 | 5000 | 1703 |
| | P | | 174.10 | 51.98 | 262.77 | 60.61 |
| | Pi | | 70.66 | 17.33 | 126.27 | 32.94 |

### *6.4.6 Modelling and forecasting the runoff*

Based on the identified and reconstructed chaotic dynamics of both runoff and rainfall, an attempt was made to build short-term forecasting models utilising chaos theory and the local modelling approach elaborated in Section 3.3.7. Initially, univariate local models, using only information from the discharge times series, were constructed. This analysis was extended with multivariate local models in the reconstructed phase-space, incorporating additional rainfall information for the local models. Finally, the hybrid modelling framework, mixture of local models—elaborated in Chapter 5, has again demonstrated the best forecasting performances. The runoff forecasts were further compared with those of artificial neural network models previously applied by Chuanbao (2001) in the framework of this research. Herewith we summarise the prediction results for the runoff (discharges) at the target station 51012.

#### RUNOFF PREDICTION USING UNIVARIATE LOCAL MODELS

Initially, univariate local models (linear and polynomial) were used in the reconstructed phase space of the runoff for the target station 51012 to forecast future runoff. In this experiment only information from the daily discharge time series at station 51012 was

used to build the local models. Sensitivity of the choice of the local approximation, the embedding dimension *(m)*, the time delay *(τ)* and the number of neghbours *(k)* was also investigated in order to find the optimal model parameters. The optimal parameters are those which yield the least prediction error according to certain performance criteria i.e. highest correlation coefficient or the lowest normalised root mean squared error (NRMSE) between the observed and predicted data using the test data set. The first 6499 daily discharge time series data were used as a training set and the data ranging from 6500 to 7230 samples (2 years period) were used as a test set. The range of the model parameters was obtained by reconstructing the runoff dynamics. The predictive performances of a 3rd order local polynomial models based on different model parameters ($m$, $\tau$ and $k$) are presented in Figure 6.4.22, which suggest optimal values for the embedding dimension of $m$=5, a time delay $\tau$=5 and the number of neighbours $k$=20, respectively.

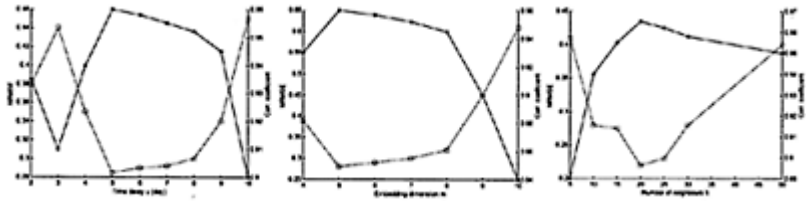


Figure 6.4.22. Performance of the local polynomial models for 1 day ahead runoff prediction at the target station 51012 for different values of the parameters $\tau$, $m$ and $k$. The line with squares represents the NRMSE and the line with stars represents the correlation coefficient between the observed and predicted runoff.

Figure 6.4.23 shows the runoff prediction for a prediction horizon of $T$=1 day ahead at the target station 51012 using local 3rd order local polynomial models with the optimal values of the parameters i.e. $\tau$=5, $m$=5 and $k$=20. Figure 6.4.24 further shows the scatter plot of the predicted and observed runoffs at the target station 51012.

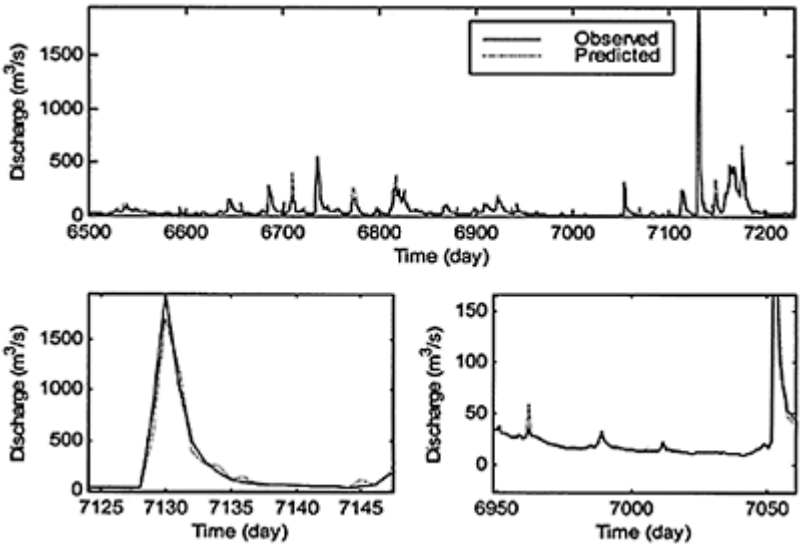



Figure 6.4.23. Prediction of the runoff at the target station 51012 using univariate local models. The prediction horizon is $T$=1 day ahead. The lower figures represent parts of the testing data set zoomed at the peak discharges and the base flow respectively.

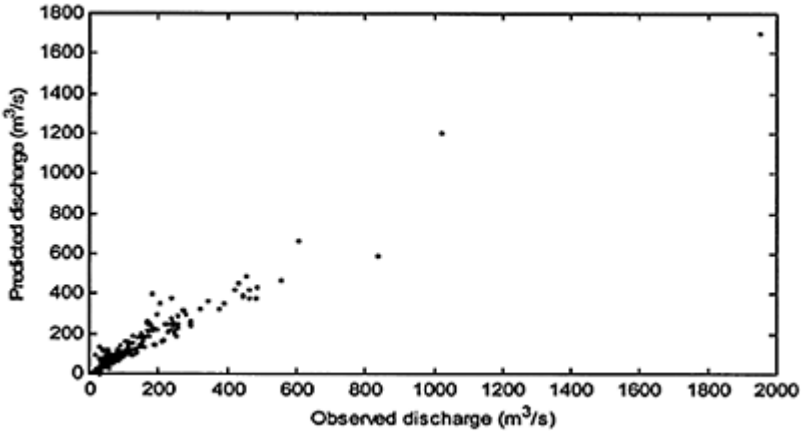



Figure 6.4.24. Scatter plot of the predicted and the observed runoff (discharge) at the target station 51012.

The results show that the overall performance of the univariate local models for a prediction horizon of $T$=1 day ahead is quite satisfactory, both for peak flows as well as for base flows. However the maximum discharge observed in the testing data set of 1950 ($m^3/s$) is underestimated by 244 ($m^3/s$). The runoff predictions using the univariate local models for a prediction horizon of T=2 days ahead show deteriorating performances (see Figure 6.4.25) as already indicated by the largest Lyapunov exponent. Table 6.4.5 summarises the runoff prediction performances for the testing data set using two prediction horizons. The performance indicators were calculated focusing on three different parts of the hydrograph: the peak flow, the base flow and the transitional flow (the rising and detention parts of the hydrograph).

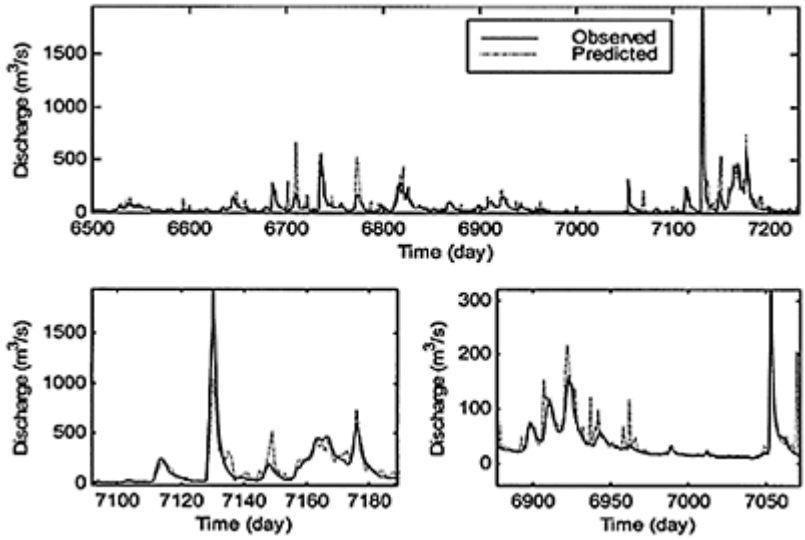



Figure 6.4.25. Prediction of the runoff at the target station 51012 using univariate local models. The prediction horizon is $T$=2 days ahead. The lower figures represent parts of the testing data set zoomed at the peak discharges and the base flow respectively.

Table 6.4.5. Summary of the performance indicators (on testing set) for the 3rd order local polynomial models for forecast horizons of $T$=1 and $T$=2 days (using $\tau$=5, $m$=5, $k$=20).

| Runoff mechanism | Performance indicators | $T$=1 day | $T$=2 days |
|---|---|---|---|
| Overall errors on testing set | MSE | 685.60 | 3830.96 |
| | NMSE | 0.0536 | 0.2995 |
| | RMSE | 26.18 | 61.89 |
| | NRMSE | 0.2315 | 0.5472 |
| | r | 0.953 | 0.843 |
| | D | 0.908 | 0.710 |
| Peaks | MAE | 163.08 | 358.76 |
| | AE on Qmax | 243.91 | 898.00 |
| Base flows | MSE | 112.96 | 535.20 |
| | NMSE | 1.1044 | 5.2326 |
| | RMSE | 10.63 | 23.13 |
| | NRMSE | 1.0509 | 2.2875 |
| | r | 0.763 | 0.571 |
| | D | 0.581 | 0.326 |
| Transitional flows | MSE | 1246.81 | 6377.88 |
| | NMSE | 0.1320 | 0.6752 |
| | RMSE | 35.31 | 79.86 |
| | NRMSE | 0.3663 | 0.8217 |
| | r | 0.940 | 0.747 |
| | D | 0.884 | 0.0.558 |

The following Figure 6.4.26 shows the accumulated volume of the runoff for the observed and predicted times series. The results indicate a clear overestimation of the volume of runoff especially for the base flows.

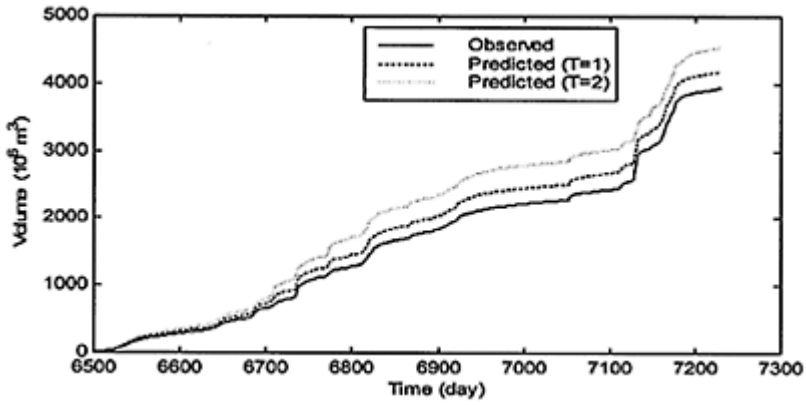


Figure 6.4.26. Comparison of the accumulated runoff volumes for the predicted and observed testing set at the target station 51012.

This overestimation of the base flow is accounted for by the fact that the model parameters were chosen by minimising the overall error between the predicted and observed runoff, which may not be necessarily the same for the different mechanisms of the runoff generation process. If we minimise the errors on the peak runoff (discharge>500 $m^3/s$), the optimal model parameters with respect to the forecasting performance are $\tau$=4, $m$=7 and $k$=8. Table 6.4.6 summarises the optimal parameters for the univariate local polynomial models with respect to the forecasting performance, focusing on the different parts of the generated runoff.

Table 6.4.6. Comparison between the model parameters ($\tau$, $m$, $k$) based on the best runoff forecasting performances using univariate local polynomial models.

| Runoff mechanism | Parameters ($\tau$, $m$, $k$) | Performance indicators | Value | Comparison with testing set (5, 5, 20) |
|---|---|---|---|---|
| Peak flow | (4, 7, 8) | MAE | 138.90 | 163.08 |
| | | AE on Qmax | 13.83 | 243.91 |
| Base flow | (7, 5, 35) | MSE | 58.93 | 112.96 |
| | | NMSE | 0.5762 | 1.1044 |
| | | RMSE | 7.68 | 10.63 |
| | | NRMSE | 0.7592 | 1.0509 |
| | | R | 0.825 | 0.763 |
| | | D | 0.681 | 0.581 |
| Transitional flow | (5, 5, 25) | MSE | 1048.53 | 1246.81 |
| | | NMSE | 0.1110 | 0.1320 |
| | | RMSE | 32.38 | 35.31 |
| | | NRMSE | 0.3332 | 0.3663 |
| | | r | 0.949 | 0.94 |
| | | D | 0.901 | 0.884 |

The results presented in the Table 6.4.6 indicate different optimal parameters for the different runoff mechanisms. The mean absolute error (MAE) on the peak flows is reduced by 15% while the absolute error on the maximal discharge ($Q_{max}$) is reduced by almost 90%. Similarly, in the base flows RMSE is reduced by 28% whereas in the transitional flows RMSE is reduced up to 8%. In terms of nonlinear dynamics, these results indicate the existence of different sub-regions of the attractor in the reconstructed

phase-space, which can be better mapped by local models using different local dimensions and neighbourhoods.

## RUNOFF PREDICTION USING MULTIVARIATE LOCAL MODELS

The univariate local models presented in the previous section do not use the rainfall as a forcing term, but generate the forecasts on the basis of the observed runoff and its reconstructed dynamics only. It may be argued that in this way a valuable source of information is ignored and, due to the possible existence of different dynamical regimes, it may not be sufficient to reconstruct the runoff dynamics using the runoff times series only. In this respect, multivariate local models incorporating information on the rainfall dynamics were tested further with a main objective of improving the runoff forecasting.

The multivariate phase-space reconstruction of the runoff dynamics using daily time series data for the runoff at the target station 51012 and the mean rainfall was solved technically using the proposed methodology described in Section 3.3.8. The optimal reconstructed multivariate phase-space can be denoted as:

$$\mathbf{Y} = \left\{ q_t^{51012}, q_{t-5}^{51012}, q_{t-10}^{51012}, q_{t-15}^{51012}, q_{t-20}^{51012}, p_t^m, p_{t-6}^m, p_{t-12}^m, p_{t-18}^m, p_{t-24}^m, \ldots, p_{t-48}^m \right\} \tag{6.5}$$

where the values of the time delays for the runoff and the mean rainfall are $\tau_q$=5 and $\tau_p$=6 days respectively, and the values of the embedding dimensions are $m_q$=5 and $m_p$=9 respectively. The optimal number of neighbours for the local models in the reconstructed multivariate phase-space was assessed by minimising the one-step ahead runoff prediction for the training (cross-validation) data set, which resulted in $k$=15. Both, local linear and local polynomial models were tested in the phase-space for the runoff prediction. The results showed that the multivariate local linear models are more robust and less sensitive to the model parameters than the local polynomial models, and were further used to forecast the runoff at the target station 51012.

The following Figure 6.4.27 and Figure 6.4.28 show the results from the runoff prediction for a prediction horizon of $T$=1 day ahead using multivariate local linear models incorporating the rainfall dynamics. In comparison with the univariate local models, the results clearly demonstrate that the multivariate local models can better capture the peak flows as well as the base flows with higher accuracy. For the maximum discharge observed in this testing data set of 1950 ($m^3/s$) the underestimation is about 45 ($m^3/s$) in contrast with the peak discharge underestimation of 244 ($m^3/s$) using univariate models.

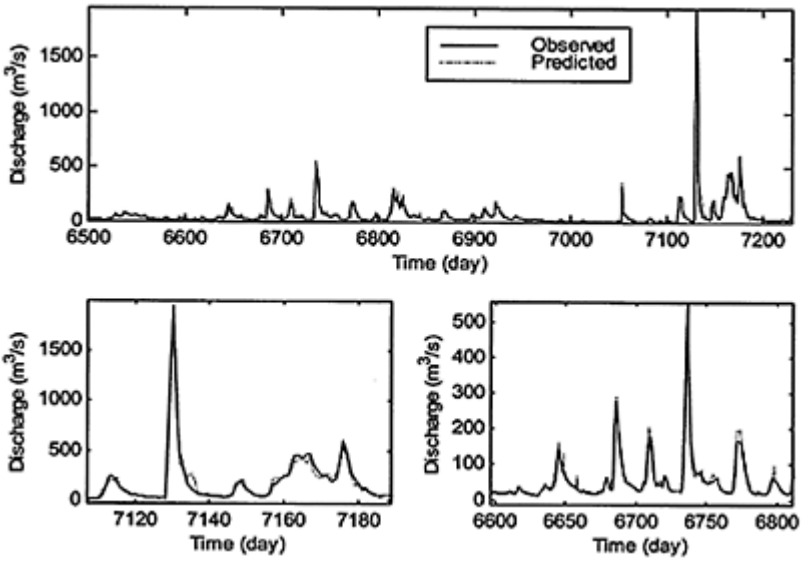



Figure 6.4.27. Prediction of the runoff at the target station 51012 using multivariate local linear models. The prediction horizon is $T$=1 day ahead. The lower figures represent parts of the testing data set zoomed at the peak discharges and the base flow respectively.

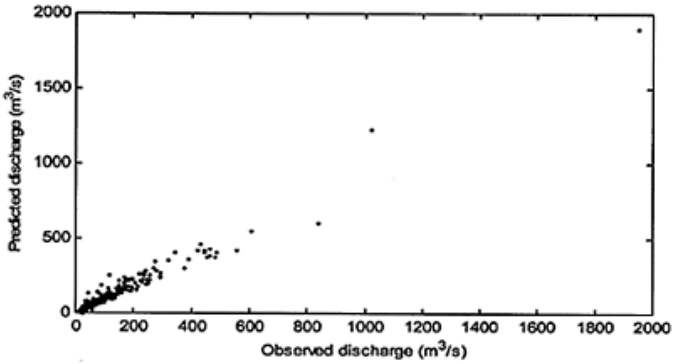



Figure 6.4.28. Scatter plot of the predicted and the observed runoff (discharge) at the target station 51012.

The prediction horizon is $T$=1 day ahead.

Figure 6.4.29 and Figure 6.4.30 show further the results from the runoff prediction at the target station 51012 for a prediction horizon of $T$=2 days ahead using multivariate local linear models. Although the multivariate local models show better predictive performances than the univariate local models, the runoff forecasts for 2 days ahead are nevertheless not satisfactory, indicating the difficulty for the runoff prediction using longer prediction horizons due to the underlying deterministic chaotic dynamics.

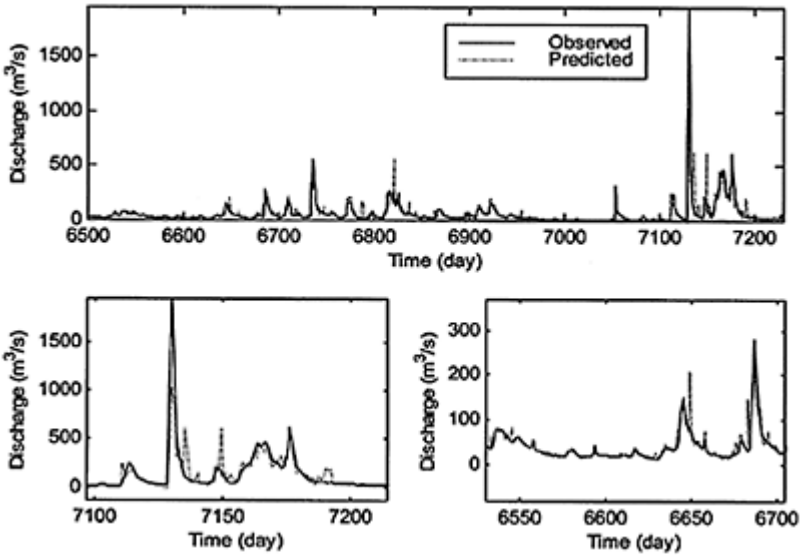



Figure 6.4.29. Prediction of the runoff at the target station 51012 using multivariate local linear models. The prediction horizon is $T$=2 days ahead. The lower figures represent parts of the testing data set zoomed at the peak discharges and the base flow respectively.

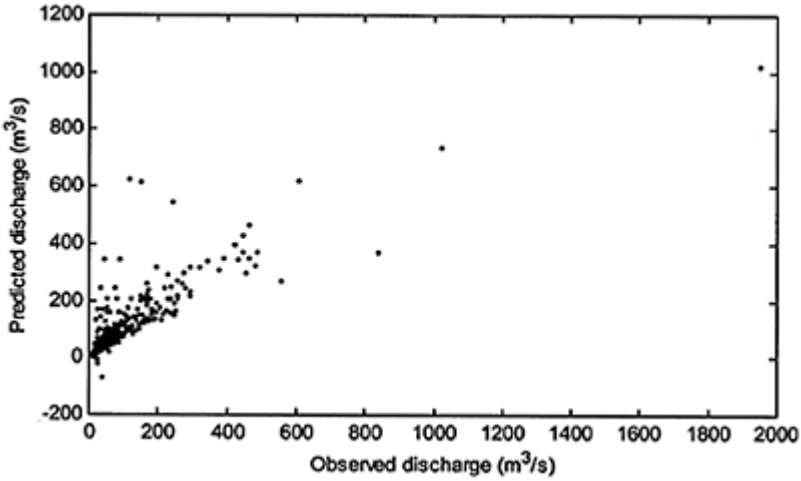



Figure 6.4.30. Scatter plot of the predicted and the observed runoff (discharge) at the target station 51012. The prediction horizon is $T$=2 days ahead.

Figure 6.4.31 shows the accumulated volume of the runoff for the observed and predicted times series using multivariate local models for both prediction horizons. The results indicate that by incorporating the runoff dynamics in the multivariate local models the overall runoff volume is estimated better. However there is a continuous small overestimation of the volume of runoff, especially in the base flows. This indicates that in the reconstructed runoff dynamics based on the time series of the runoff and rainfall, certain runoff generation processes are missing, such as the subsurface flow and groundwater flow. However, from a point of view of flood forecasting the multivariate local models are able to capture the underlying relationships between the rainfall and the runoff for a short prediction horizon. Table 6.4.7 summarises the performance indicators of the multivariate local models for runoff forecasting at the target station 51012.

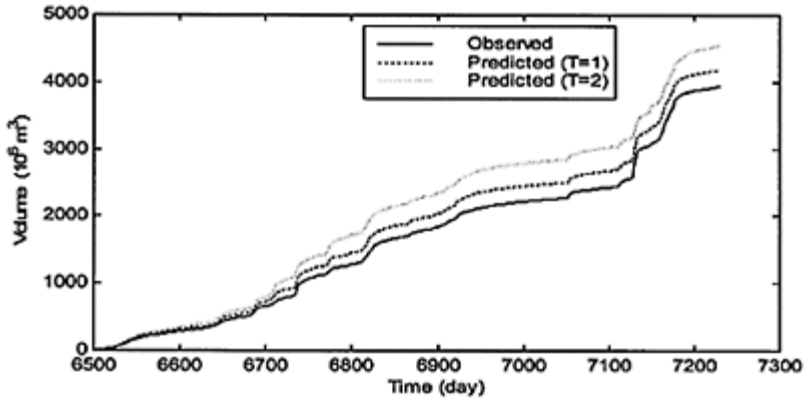



Figure 6.4.31. Comparison of the accumulated runoff volumes for the predicted (using multivariate local linear models) and observed testing set at the target station 51012.

Table 6.4.7. Summary of the runoff forecasting performance indicators (on testing set) using multivariate local linear models for forecast horizons of $T$=1 and $T$=2 days.

| Flow mechanism | Performance indicators | $T$=1 day | $T$=2 days |
|---|---|---|---|
| Overall errors on testing set | MSE | 414.84 | 3452.55 |
| | NMSE | 0.0324 | 0.2699 |
| | RMSE | 20.36 | 58.76 |
| | NRMSE | 0.18 | 0.5195 |
| | r | 0.984 | 0.856 |
| | D | 0.9682 | 0.732 |
| Peaks | MAE | 132.404 | 392.47 |
| | AE on $Q_{max}$ | 45.38 | 923.10 |
| Base flows | MSE | 43.26 | 608.27 |
| | NMSE | 0.4229 | 5.940 |
| | RMSE | 6.5772 | 24.66 |
| | NRMSE | 0.6503 | 2.4372 |

| | | | |
|---|---|---|---|
| | r | 0.859 | 0.488 |
| | D | 0.7378 | 0.238 |
| Transitional flows | MSE | 731.47 | 4390.36 |
| | NMSE | 0.0774 | 0.4648 |
| | RMSE | 27.04 | 66.25 |
| | NRMSE | 0.278 | 0.6817 |
| | r | 0.961 | 0.783 |
| | D | 0.923 | 0.614 |

In the previous multivariate experiments, the average daily rainfall over the 17 rainfall stations was used in the phase-space reconstruction of the system. The spatial distribution of the rainfall was thus not incorporated in the runoff dynamics. In order to investigate the influence of the rainfall spatial distribution on the runoff dynamics, the average rainfall from the 5 HRUs mentioned in Section 6.4.3 were used in the following experiment as an analogy for the extent of the spatial information incorporated into the multivariate local linear models. The embedding dimension for each HRU was taken as $m$=9, while the time delay was varied between $\tau$=2 and $\tau$=6 days in order to achieve the best predictive performances for the testing data set. The results from the runoff forecasting for a prediction horizon of $T$=1 day ahead using multivariate local linear models and incorporating the spatial distribution of the rainfall over the 5 HRUs are summarised in Table 6.4.8.

Table 6.4.8. Summary of the runoff forecasting performance indicators using multivariate local linear models incorporating spatial rainfall distribution for forecast horizon of $T$=1 day.

| Flow mechanism | Performance indicators | $T$=1day |
|---|---|---|
| Overall errors on testing set | MSE | 516.40 |
| | NMSE | 0.0403 |
| | RMSE | 22.72 |
| | NRMSE | 0.2007 |
| | r | 0.9796 |
| | D | 0.9596 |
| Peaks | MAE | 130.22 |
| | AE on Qmax | 43.26 |
| Base flows | MSE | 125.58 |
| | NMSE | 1.2277 |
| | RMSE | 11.20 |
| | NRMSE | 1.1080 |

| | | |
|---|---|---|
| | r | 0.720 |
| | D | 0.5184 |
| Transitional flows | MSE | 877.16 |
| | NMSE | 0.0929 |
| | RMSE | 26.61 |
| | NRMSE | 0.3047 |
| | r | 0.953 |
| | D | 0.9082 |

By comparing the performance indicators in Table 6.4.8 and in Table 6.4.7, the incorporation of the spatial rainfall distribution does not improve the overall results of the runoff forecasts at the target station 51012. The overall performance indicators (errors) show even a slightly smaller accuracy in terms of correlation coefficient between the predicted and the observed runoff. However, there is a slight improvement in the peak flow prediction using the spatial runoff distribution. Another interesting finding is that although the performance indicators of the base flows show lower correlation coefficient ($r$=0.72), due to oscillations of the predicted values around the measured base flow, the accumulated runoff volume demonstrates that the multivariate local models incorporating the spatial rainfall distribution are able to maintain a good representation of the overall water balance. This is presented in Figure 6.4.32.

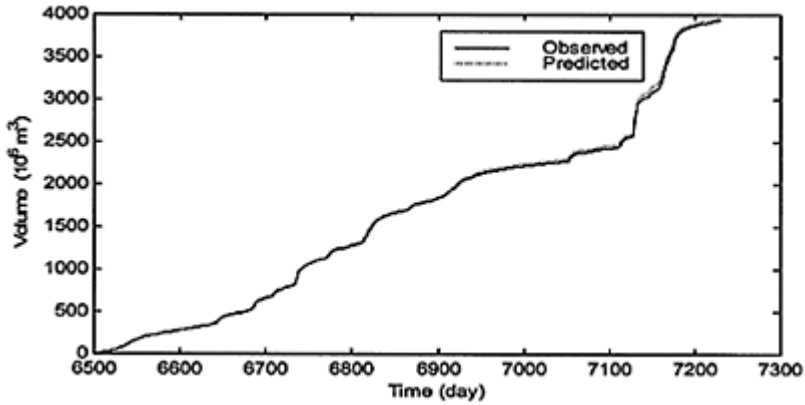



Figure 6.4.32. Comparison of the accumulated runoff volumes for the predicted (using multivariate local linear models incorporating spatial rainfall distribution) and observed testing set at the target station 51012. The prediction horizon is $T$=1 day ahead.

## COMPARISON OF THE RUNOFF PREDICTION WITH ARTIFICIAL NEURAL NETWORKS

In the framework of this work, Chuanbao (2001) investigated different ANN architectures in order to forecast runoff at the target station 51012 from the rainfall. In that modelling experiment the whole data set was divided into three parts for training, cross validation and testing, respectively. The data from 1976 to 1992 were selected to train the network, the data from 1933 to 1994 were used for cross validation, and finally the data from 1995 to 1996 were chosen to test the neural networks. Three different types of neural network architectures were tested, namely: (i) multi-layered perceptrons (MLP); (ii) recurrent networks and (iii) modular networks. The best runoff forecasting performances were demonstrated by the modular neural network model using spatially distributed rainfall from the 5 HRUs.

In the previous experiments using both the univariate and the multivariate local models, the testing data was selected within the range of 6500 to 7230 data samples. In order to make the performance measures comparable with the ANN results, the testing data was extended from 7000 to 7671 (year 1995 and 1996) and both the univariate and the multivariate local models were tested for runoff prediction. This testing data set contains runoff peaks in the range of 4000 ($m^3/s$), which were not previously introduced to the local models while optimising their predictive performances, in order to test the extrapolation capabilities of the local models. The results from the runoff prediction at the target station 51012 using multivariate local linear models for a prediction horizon of $T$=1 day ahead are shown in Figure 6.4.33 with a scatter plot in Figure 6.4.34. The results indicate that although the predicted and observed hydrographs are in a good agreement, the extreme runoff peaks are still underestimated. The maximum observed peak runoff of 3750 ($m^3/s$) is underestimated by about 820 ($m^3/s$). Table 6.4.9 summarises the comparison of the runoff forecasts between the univariate and multivariate local models and the best neural network model. The performance indicators show that the runoff forecasts using a simple nonlinear forecasting technique based on the univariate local models in the reconstructed runoff phase-space are comparable with the ANN model. Multivariate local linear models incorporating the rainfall dynamics clearly outperform the modular ANN model. In particular, multivariate LLMs give better results in terms of the peak discharge forecast.

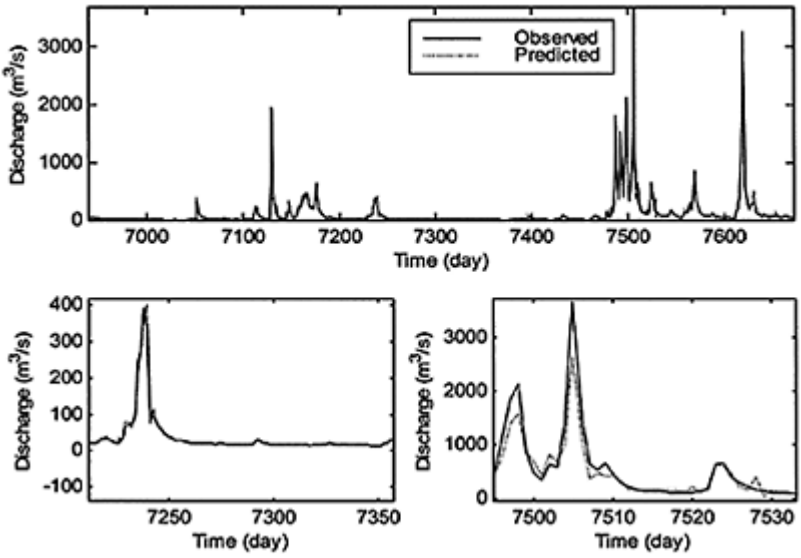



Figure 6.4.33. Prediction of the runoff at the target station 51012 using multivariate local linear models. The prediction horizon is $T$=1 day ahead. The lower figures represent parts of the testing data set zoomed at the peak discharges and the base flow respectively.

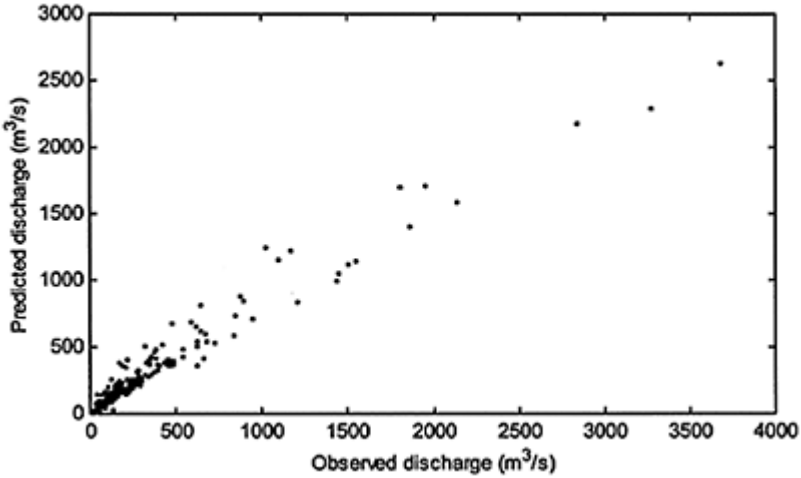



Figure 6.4.34. Scatter plot of the predicted and the observed runoff (discharge) at the target station 51012. The prediction horizon is $T$=1 day ahead.

Table 6.4.9. Comparison of the univariate and multivariate local models with ANN and Naïve models for the runoff forecasts at the target station 51012. The testing data set ranges from 7000 to 7671 data samples with a prediction horizon of $T$=1 day ahead.

| Performance indicators | Modular ANN model | Naïve model | Univariate local 3rd order polynomial models | Multivariate local linear models |
|---|---|---|---|---|
| MSE | 10722 | 43026 | 10994 | 6508.05 |
| RMSE | 103.5 | 207.7 | 104.8 | 80.67 |
| **r** | **0.9631** | **0.7707** | **0.9585** | **0.9801** |
| D | 0.9276 | 0.5939 | 0.9187 | 0.9606 |

## RUNOFF PREDICTION USING MIXTURE OF MODELS (HMMM)

The final experiment performed in this case study is the application of the mixture of models framework elaborated in Chapter 5. The hydrograph analysis performed in Section 6.4.3 has shown the existence of different dynamic regimes in the generation of runoff. Furthermore, the local modelling experiments using both, univariate and multivariate local models showed that there are clear regions in the attractor of the reconstructed phase-space that can be modelled using different local models with different parameters ($\tau$, $m$ and $k$), i.e. capacity. This knowledge was used further to setup a mixture of multivariate local linear models whereby their activation functions (gating) are modelled by a Hidden Markov process.

The multivariate local linear models (experts), based on the reconstructed phase-space from the time series of the runoff and the mean rainfall (Eq. 6.5) are used to model each of the hidden states (dynamic regimes) of the system. For the learning of the activation function of each expert (model) the training set consisted of 6999 samples (more than 18 years of daily data). Part of this training set was used as a cross-validation set to determine the number of the local models, each representing a possible dynamic regime of the system. Various combinations of the parameters of the local models, such as the time delay, embedding dimension and the number of the nearest neighbours, were investigated. The same test set of the last 2 years of the data (from 7000 to 7671 data samples) was used to evaluate and compare the performance of the HMMMs. Table 6.4.10 and Figure 6.4.35 summarise the main results of the mixture of models framework on the testing data set for the best combination of the local models.

Table 6.4.10. Summary of the runoff forecasting performance indicators using mixture of multivariate local models for forecast horizon of $T$=1 day. The testing data set ranges from 7000 to 7671 data samples.

| Runoff mechanism | Performance indicators | $T$=1 day |
|---|---|---|
| Overall errors on testing set | MSE | 3045.0 |
| | RMSE | 55.18 |
| | **r** | **0.9858** |
| Peaks | MAE | 181.40 |
| | AE on Qmax | 99.90 |
| Base flows | MSE | 30.86 |
| | RMSE | 5.55 |
| | r | 0.9023 |
| Transitional flows | MSE | 1089.0 |
| | RMSE | 33.0 |
| | r | 0.9642 |

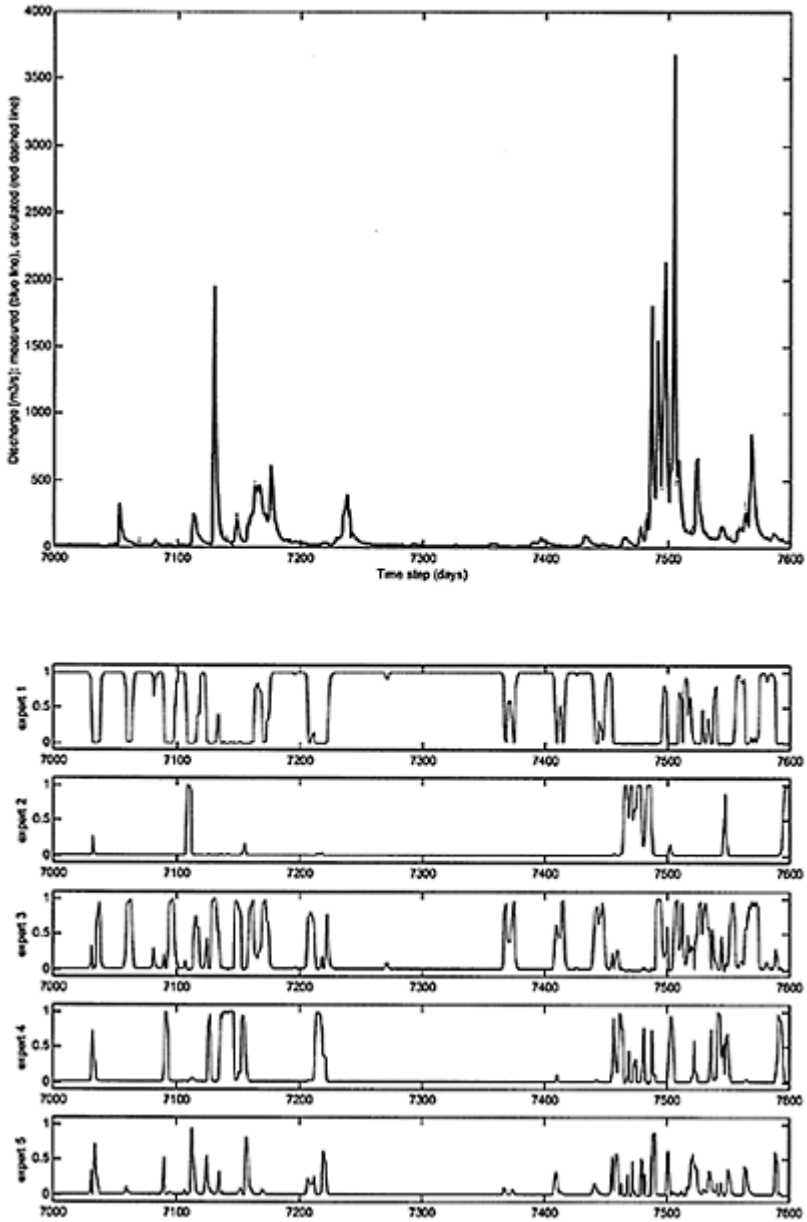



Figure 6.4.35. Prediction of the runoff at the target station 51012 using mixture of multivariate local linear models. The prediction horizon is $T$=1 day ahead. The lower figure shows the

activation functions of each of the models (experts). Soft combination of each model's prediction is used to generate the runoff prediction.

The results indicate that the mixture of models framework improved the prediction of both, the base flow and the peak discharges. The underestimation in the prediction of the maximal recorded peak runoff of 3750 ($m^3/s$) is significantly reduced to about 100 ($m^3/s$).

The combination of five different multivariate local models gave the best predictive performances. From the Figure 6.4.35 is noticeable that the expert 1 specialises on the prediction of the base flow (no impulse in the dynamical system from the rainfall) while the other four experts (models) specialise on the different runoff generation regimes. The parameters of the expert 1 used in the HMMEs (time delay $\tau$=7, embedding dimension $m$=5 and number of nearest neighbours $k$=45 for the discharge time series) differ significantly from the parameter of the expert 3 ($\tau$=4, $m$=7 and $k$=5 for the discharge time series), which on the other hand specialises on modelling of peak runoffs. This increase of the embedding dimension implies that the number of the essential state variables governing the dynamics of the runoff during extreme events is different compared to the number of the essential variables necessary to model the generation of the base flow.

### *6.4.7 Summary and conclusions*

In this case study we have demonstrated the application of the methods based on the theory of nonlinear dynamics and chaos to rainfall-runoff modelling for the Xixian catchment of Huai River basin. As the runoff generation process is highly nonlinear, time varying and spatially distributed, the underlying dynamics of the system were investigated using multivariate phase-space reconstruction techniques. The results provide evidence that both the runoff dynamics and the rainfall dynamics can be characterised by deterministic chaos. The correlation integral analysis together with the Lyapunov exponents method demonstrated the existence of a strange attractor ($d_c$=3.2) and a hyper-chaotic behaviour of the runoff dynamics. The stability analysis based on the Lyapunov exponents for the long-term behaviour of the runoff dynamics showed that the average rate of divergence of the small perturbations in the runoff dynamics dominate the average rate of their convergence. This implies that the trajectory of the runoff dynamics in the reconstructed phase-space is not bounded, indicating that the system may not be asymptotically stable exhibiting different dynamical regimes. This finding was further supported by an analysis of selected runoff hydrographs that showed the existence of possibly four different dynamic regimes in generating the runoff based on the available data for the period of 21 years from 1976 to 1996.

Based on the identified and reconstructed chaotic dynamics of both the runoff and the rainfall, short-term forecasting models (one day ahead) utilising the local modelling approach were constructed in order to predict the runoff at the target station 51012. The univariate local models, tested initially using only information from the discharge times series showed relatively good predictive performance comparable with a modular multivariate neural network model; see Table 6.4.9. The short-term forecasting runoff models were further extended with multivariate local models in the reconstructed

multivariate phase-space incorporating additional rainfall information. Finally, the hybrid modelling framework—mixture of local models—has demonstrated the best forecasting performances, with an overall root mean squared error of $RMSE$=55.2 ($m^3/s$) and a correlation coefficient of $r$=0.9858 between the observed and the predicted runoffs for the test data set that includes the highest observed discharge peaks.

In summary, in this case study we have demonstrated that the methodology based on the theory of nonlinear dynamics and chaos supported by the mixture of modelling framework can serve as an efficient tool for building accurate short-term rainfall-runoff models, especially in real-time flood forecasting and management.

# Chapter 7
# **Conclusions**

A hydroinformatics system represents an electronic knowledge encapsulator that models part of the real world and can be used for the simulation and analysis of physical, chemical and biological processes in water systems, for a better management of the aquatic environment. Thus, modelling is at the heart of hydroinformatics. The theory of nonlinear dynamics and chaos and the extent to which recent improvements in the understanding of inherently nonlinear natural processes present challenges to the use of mathematical models in the analysis of water and environmental systems are elaborated in this work. In particular, we demonstrate that the deterministic chaos present in many nonlinear systems can impose fundamental limitations on our ability to predict natural processes even when well-defined mathematical models exist. On the other hand, the methodologies and tools based on the theory of nonlinear dynamics and chaos elaborated in this work can provide means for a better accuracy of short-term predictions as demonstrated through the practical applications.

In Chapter 3, we described, elaborated mathematically and illustrated the main concepts of the theory of nonlinear dynamics and deterministic chaos. We further introduced and demonstrated the methods and techniques for the identification, reconstruction, delineation and quantification of the underlying dynamics of nonlinear dynamical systems from a time series of observables. The phase-space reconstruction based on a univariate time series was further extended and elaborated using the multivariate embedding methodology proposed in this work. It was elaborated further how models can be “learned” from data that realistically map the underlying structure dictating the dynamical evolution of the system. This modeling approach is closely connected to data-driven modelling based on the computational intelligence, search and optimisation methods addressed in Chaper 2.

From a modelling standpoint, irregularity and chaos are fundamental to nonlinear dynamical systems, which even with a few variables can generate very rich and complex dynamical structures. These are characterised by the presence of chaotic dynamics, different dynamical regimes (even coexisting attractors) and an irregular dynamical evolution between them. From a modelling perspective such complexities were addressed by the development of a novel hybrid framework that draws on both chaos theory and dynamic Bayesian networks. This modelling framework, elaborated in Chapter 5, combines the multivariate phase-space reconstruction of the underlying dynamics based on a time series of observables and a mixture of local models learned in a dynamic Bayesian network formalism represented through a hidden Markov model.

In Chapter 6, the proposed modelling framework was applied for identification, modelling and prediction of hydrodynamical and hydrological systems: sea water level and surge dynamics along the Dutch cost, precipitation dynamics at the De Bilt

meteorological station in The Netherlands and rainfall-runoff dynamics of the Huai river in China. The main results from these practical applications are summarized as follows:

*Case study 1: Nonlinear dynamics, chaos and predictability of the water levels and surges along the Dutch coast*

Based on a nonlinear analysis, phase-space reconstruction and estimation of various geometrical and dynamical invariants, the dynamics of both water levels and surges along the Dutch coast can be characterised as *deterministic chaos*. The presence of the chaotic dynamics together with the positive Lyapunov exponents implies that there are limits of predictability for any model (refer to Table 6.2.2 and Table 6.2.3). However, reliable short-term predictions are possible.

The chaotic behaviour occurs because water levels and surges, including astronomical contributions and the contributions from the meteorological forcing, are the result of a complex, coupled nonlinear dynamical system. The analysis of the shallow-water dynamics demonstrated and explained the appearance of the double low water and the distortion of the duration of the high waters.

The Lyapunov exponents and the entropies have significant consequences for numerical models that are based on solutions of the hydrodynamic equations of motion. The implication of the presence of deterministic chaos in surge dynamics is that estimates of future behaviour are very sensitive to mathematical formulations and assumptions, the choice of various coefficients and parametrisation. The system's current state may also be inadequately modelled or measured. The main implication however is that improvements in forecasting may require significant improvements in the accuracy of the numerical solution of the mathematical terms, coefficients and measurements, which are used as initial and boundary conditions, especially in the meteorological forcing. Data assimilation techniques, based on very accurate measured data may help to improve the prediction performances.

Taking into account the presence of deterministic chaos in the water level and surge dynamics, a mixture of multivariate adaptive local modelling in the reconstructed phase-space of the dynamical system, which uses information from the real dynamical neighbours, has demonstrated good capability for reliable short-term predictions. For the Hoek van Holland location, the overall prediction error for the surge 10 hours ahead is about 10.5cm. For stormy sea dynamics the prediction error is about 12 (cm) and about 8 (cm) for non-stormy sea dynamics (the test data set was taken from the period between 1.01.95–31.08.95).

The identification and selection of proper dynamical neighbours from historical time series data are the key issues in the local modelling approach adopted in this work. The dynamical selection of the types and number of neighbours in the modelling procedure indicates that there are different dynamical regimes present in the sea dynamics that may be modelled using different alternative types of models (e.g. local models, neural networks, etc.). In this study the mixture of models framework showed the best predictive performances.

Local uncertainty analysis is an appropriate technique for studying the predictability of the surge dynamics. Although the overall predictability is about 80%, there exist certain dynamical situations when the predictability is much better than the average predictability and certain dynamical situations when the predictability is quite low, especially for negative surges.

Chaos theory can serve as an efficient tool for accurate and reliable short-term predictions of water levels in order to support decision-makers in ship navigation.

*Case study 2: Chaos in rainfall dynamics*

In this application we investigated the existence of chaos in rainfall dynamics using methods and techniques from nonlinear dynamics and chaos mathematics, based on the rainfall time series recorded at the De Bilt meteo station in the Netherlands. The main question of the existence of structurally different chaotic dynamics in the rainfall using different temporal scales of the observables was addressed by the analysis of 15min, hourly, daily and weekly rainfall data.

The correlation dimension method provided evidence of the existence of a low-dimensional attractor for the different rainfall data sets aggregated over different time periods, thus suggesting the existence of chaotic dynamics. Based on the attractor dimensions that were generated for the 15min, hourly, daily and weekly rainfall data, the minimum number of variables essential to model the rainfall dynamics was identified as 3, 4, 11 and 11, respectively. The indicative number of sufficient variables to fully describe the rainfall dynamics on different temporal scales was identified as 40, 38, 30 and 11, respectively. The effects of the time delay value, used for the phase-space reconstruction, on the attractor dimension estimation were also investigated in order to compare the results obtained from the average mutual information function.

The Lyapunov exponents computed on the 15min, hourly, daily and weekly rainfall data, demonstrated strong evidence of the existence of chaotic dynamics in the 15min and hourly data and hyper-chaos in the daily and weekly rainfall dynamics. The existence of positive Lyapunov exponents for all the rainfall data sets clearly showed the limits of the predictability of any model.

The method of surrogate data for distinguishing between chaotic and stochastic rainfall dynamics based on the continuous wavelet transform, together with the test for nonlinearity, provided evidence that the rainfall dynamics is different from a linear stochastic process. In addition, the simple nonlinear noise-reduction algorithm applied to the different rainfall data sets improved the results for the correlation dimension estimation, and thus the reconstructed phase-space.

The nonlinear prediction method based on univariate local modelling in the reconstructed phase-space enabled us to check the prediction accuracy using different time horizons and with respect to: (i) number of neighbours; (ii) optimal time delay; and (iii) the embedded dimension. The results indicated a reasonable short-term predictability for the hourly and daily rainfall, but a sharp drop in the prediction accuracy due to the presence of hyperchaotic dynamics. The mixture of models framework, elaborated in Chapter 5, using a different capacity for the models (experts), showed the best predictive performances.

In summary, the results from this application lead to the conclusion that structurally different chaotic dynamics in the rainfall exist at different temporal scales. However, rainfall is a multidimensional spatio-temporal phenomenon. The rainfall dynamics are not only highly fluctuating in time but also in space. These spatio-temporal signatures (patterns) are not independent but rather dependent. In addition, they usually occur at rather small grid resolutions, such as 5–10 km. Much more needs to be done in the collection of fine-resolution data in space and time in order to be able to study the spatiotemporal rainfall dynamics and to improve the numerical weather forecasting

models. The recent advances in remote sensing and radar surveillance technology will help in the collection of such kind of data. At present it is not clear whether the dynamics of spatiotemporal rainfall patterns can be described by an attractor in a phase-space over a certain area, which in turn may improve the short-term rainfall predictions.

*Case study 3: Rainfall-runoff modelling*

In this case study we demonstrated the application of the methods based on the theory of nonlinear dynamics and chaos to rainfall-runoff modelling for the Xixian catchment of Huai River basin. As the runoff generation process is highly nonlinear, time varying and spatially distributed, the underlying dynamics of the system was investigated using multivariate phase-space reconstruction techniques. The results provide evidence that both, the runoff dynamics and the rainfall dynamics can be characterised as deterministic chaos. The correlation integral analysis together with the Lyapunov exponents method demonstrated the existence of a strange attractor ($d_c$=3.2) and a hyper-chaotic behaviour of the runoff dynamics. The stability analysis based on the Lyapunov exponents for the long-term behaviour of the runoff dynamics showed that the average rate of divergence of the small perturbations in the runoff dynamics dominates the average rate of their convergence. This implies that the trajectory of the runoff dynamics in the reconstructed phase-space is not bounded, indicating that the system may not be asymptotically stable in that it exhibits different dynamical regimes. This finding was further supported by an analysis of selected runoff hydrographs that showed the existence of possibly four different dynamic regimes in the generation of the runoff based on the available data for the period of 21 years from 1976 to 1996.

Based on the identified and reconstructed chaotic dynamics of both the runoff and the rainfall, short-term forecasting models (1 day ahead) utilising the local modelling approach were constructed in order to predict the runoff at the target station 51012. The univariate local models, tested initially using only information from the discharge times series, showed a relatively good predictive performance comparable with a modular multivariate neural network model; see Table 6.4.9. The short-term forecasting runoff models were further extended with multivariate local models in the reconstructed multivariate phase-space incorporating additional rainfall information. Finally, the hybrid modelling framework, based on a mixture of local models, has demonstrated best forecasting performances, with an overall root mean squared error of *RMSE*=55.2 ($m^3/s$) and a correlation coefficient of $r$=0.9858 between the observed and the predicted runoffs for the testing data set.

In summary, in this case study we demonstrated that the methodology based on the theory of nonlinear dynamics and chaos supported by the mixture of modelling framework can serve as an efficient tool for building accurate short-term rainfall-runoff models, especially in real-time flood forecasting and management.

*Postscript*

The potential role of the theory of nonlinear dynamics and chaos in modeling the natural processes in our aquatic environment is generally diverse. At the simplest level, the elaborated methodology and tools can be used for nonlinear time series analysis and system dynamics identification relating various variables describing the underlying processes, which in turn can generate important knowledge necessary for the mathematical description of the system. At the other end of the spectrum, chaos theory

may be used to generate important components of physically-based mathematical modeling systems, as well as a stand-alone data-driven modeling approach, whereby a mixture of multivariate local models may be used to describe specific physical processes or the complete system dynamics. The greatest potential for the modeling approach based on chaos theory in hydroinformatics, however, might be in the area of real-time operational forecasting and control of water systems, due to the capability of fast and accurate short-term forecasting as demonstrated in this work.

Finally, the natural processes and phenomena in our aquatic environment are complex and adaptive dynamical systems exhibiting nonlinear interactions, chaos, switching dynamical regimes, adaptation, emergence and evolution. While we may always make further progress in understanding particular systems and their underlying processes, there will always be some processes that lie just beyond our scientific abilities to predict. Natural processes, being composed of all these features, will always have novelty, richness, and beauty that can never be exhausted nor fully computed. As with the surge water levels, precipitation dynamics and rainfall-runoff dynamics studied in this work, we can appreciate their splendour because we can simulate them, but only to limited accuracy. If all natural phenomena were either perfectly describable or absolutely indescribable, not only would they be uninteresting and non-challenging for the scientists, but life would be impossible.

# References


Abbott, M.B. (1991). Hydroinformatics: Information Technology and the Aquatic Environment, Aldershot, UK/Brookfield, USA: Ashgate.

Abbott, M.B. (1994). Hydroinformatics: A Copernican revolution in hydraulics. Journal of Hydralics Research, Vol 32. Extra Issue, IAHR, pp. 3–13.

Abbott, M.B., Bathurst, J.C., Cunge, J.A., O'Connell, P.E. and Rasmussen, J., (1986). An introduction to the European Hydrological System—System Hydrologique Europeen, "SHE", 1: History and philosophy of a physically- based, distributed modelling system, Journal of Hydrology, 87:45–59.

Abbott, M, B., and Joniski A. (2001). The democratisation of decision-making processes in the water sector II. Journal of Hydroinformatics 3, pp 35–48.

Abarbanel H.D.I. (1996). Analysis of observed chaotic data. Springer.

Alvarez, G.M. (2001). Neuro-fuzzy modeling in Engineering Geology. PhD thesis, Balkema, Rotterdam.

Babovic, V. (1996), Emergence, Evolution, Intelligence; Hydro-informatics. PhD thesis, Balkema, Rotterdam.

Babovic, V. and Keijzer, M. (1999). Forecasting of River discharges in the presence of chaos and noise, Coping with floods: Lessons Learned form Recent Experiences, Marsalek, J. (ed.) Kluwer, Dodrecht.

Babovic, V., Keijzer, M. and Stefansson, M. (2001). Chaos Theory, Optimal Embedding and Evolutionary Algorithms, D2K Technical Report, Danish Hydraulic Institute.

Bakshi, B.R. and G.Stephanopoulos (1993). Learning at Multiple Resolutions: Wavelets as Basis Functions in Artificial Neural Networks and Inductive Decision Trees. In Wavelet Applications in Chemical Engineering, eds. R.Motard and B.Joseph, Kluwer Inc., Boston.

Baum, L.E. (1972). An inequality and associated maximisation technique in statistical estimation of probabilistic functions of Markov process, Inequalities 3:1–8.

Baum, L.E., Petrie, T., Soules, G. and Weiss, N. (1970). A maximisation technique occurring in the statistical analysis of probabilistic functions of Markov chains, Annals of Mathematical Statistics 41:164–171.

Bayes, T. (1958). An essay towards solving a problem in the doctrine of chances, Biometrika, 46:293–298.

Badii, R.A. and A.Politi (1997). Complexity: Hierarchical Structures and Scaling in Physics, Camb.Uni. Press.

Baum, L.E., Petrie, T., Soules, G. and Weiss, N. (1970). A maximisation technique occurring in the statistical analysis of probabilistic functions of Markov chains, Annals of Mathematical Statistics, 41:164–171.

Baxt, W.G. (1990). Use of an Artificial Neural Network for Data Analysis in Clinical Decision-Making: The Diagnosis of Acute Coronary Occlusion. Neural Computation, 2, 480–489.

Bengio, Y. and P.Frasconi (1995). An input-output HMM architecture, Advances in Neural Information Processing Systems 7:427–434, MIT Press.

Bengio, S. and Bengio Y. (1996). Input-output HHM's for sequence processing, IEEE Transactions on Neural Networks 7(5):1231–1249.

Bengio, Y. (1999). Markovian Models for Sequential Data, Neural Computing Surveys, 2:129–162.

Berger, J.O. 1985. Statistical Decision Theory and Bayesian Analysis. Springer-Verlag.

Benjamin, J.R., and Cornell, C.A. (1970). Probability, Statistics, and Decision for Civil Engineers. McGraw-Hill Book Co., New York.

Billings, S.A. (1980). Identification of nonlinear systems: A survey. IEE Proc. 127, 272–285.
Bishop, C.H. (1995). Neural Networks for Pattern Recognition, Oxford University Press, Oxford.
Blanz, V., Schölkopf, B., Bülthoff, H., Burges, C., Vapnik, V., and Vetter, T. (1996). Comparison of view—based object recognition algorithms using realistic 3D models. In: C.von der Malsburg, W.von Seelen, J.C. Vorbrüggen, and B.Sendhoff (eds.): Artificial Neural Networks—ICANN'96. Springer Lecture Notes in Computer Science Vol. 1112, Berlin, 251–256.
Boogaard, H.F.P.van den; Gerritsen, H.; Mynett, A.E. (2003). Uncertainty assessment in basis function models with application in hydraulic engineering, IAHR XXX.
Boogaard, h.F.P.van den, A.E.Mynett, H.Gerritsen (2001). Uncertainty assessment in tidal analyses, Proceedings XXIX-th IAHR Congress, Beijing, 16–21 September 2001.
Boogaard, H.F.P.van den, Brummelhuis, P.G.J.ten, Mynett, A.E. (2000). On-Line Data Assimilation in Auto-Regressive Neural Networks., Conf. Proc. Hydroinformatics 2000, Cedar Rapids, Iowa, USA.
Boogaard, H.v.d., and M.R.A.Gent (1998). Neural Network and Numerical Modelling of Forces on Vertical Structures, MAST-PROVERBS report, Delft Hydraulics.
Breiman, L. (1993). Hinging Hyperplanes for Regression, Classification, and Function Approximation. IEEE Trans. on Information Theory 39:999–1013.
Burges, C. and Vapnik, V. (1995). A new method for constructing artificial neural networks. Technical report, AT&T, Bell Laboratories, NJ.
Burges, C.J.C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. Kluwer Academic Publishers, Boston.
Cacciatore, T.W. and Nowlan S.J. (1994). Mixture of controllers for jump linear and non-linear plants, Advances in Neural Information Processing Systems 6:719–726, NIPS 1993, MIT Press.
Chen, Q., A.E.Mynett and A.W.Minns (2001). Application of cellular automata to modelling competitive growth in Lake Veluwe, J.Ecological Modelling, 147:253–265.
Chentsov, N.N. (1962). Estimation of unknown probability density based on observations. Dokl. Akad. Nauk SSSR, 147 (1962), pp. 45–48 (in Russian).
Cheeseman, P. and Stutz, J. (1995). Bayesian Classification (AutoClass): Theory and Results, Advances in Knowledge Discovery and Data Mining, Usama M.Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, & Ramasamy Uthurusamy, Eds. The AAAI Press, Menlo Park.
Chuanbao, Z.(2001). Application of ANNs to Rainfall-Runoff Modelling in the Upper Reach of the Huai River basin, M.Sc. Thesis H.H 414, IHE, Delft.
Cooper, G. (1990). Computational complexity of probabilistic inference using Bayesian belief networks, Artificial intelligence, 42:393–405.
Cooke, R M., and J.M.van Noortwijk (2000). Graphical methods. In A.Saltelli, K.Chan, E.M.Scott, editors, Sensitivity Analysis, pages 245–264. Chichester: John Wiley & Sons, 2000.
Cooke, R.M. and J.M.van Noortwijk (1999). Generic graphics for uncertainty and sensitivity analysis. In G.I.Schuëller and P.Kafka, editors, Safety and Reliability, Proceedings of ESREL '99—The Tenth European Conference on Safety and Reliability, Munich-Garching, Germany, 1999, pages 1187–1192. Rotterdam: Balkema,
Cooke, R.M. (1997). Uncertainty modeling: examples and issues. Safety Science, 26, no 1/2:49–60.
Cooke R.M. (1997). Markov and entropy properties of tree and vines- dependent variables. In Proceedings of the ASA Section of Bayesian Statistical Science.
Cooke R.M. (1991). Experts in Uncertainty. Oxford University Press, New York.
Cooke R.M., Waij R. (1986). Monte carlo sampling for generalized knowledge dependence with application to human reliability. Risk Analysis, 6:335–343.
Cooper, G. and Herskovits, E. (1992). A Bayesian method for induction of probabilistic networks from data, Machine learning, 9:309–347.
Cortes, C. and Vapnik, V. (1995). Support vector networks. Machine Learning, 20:1–25.
Cunge, J.A. (1969). On the subject of flood propagation computation method (Muskingum method),. Journal of Hydraulic Research, Vol.7, No.2, IAHR, pp. 205–230.

Cunge, J.A. and M.Erlich (1999). Hydroinformatics in 1999: What is to be done? Journal of Hydroinformatics Vol 1, pp. 21–31.
Cybenko, G. (1989). Approximation by Superposition s of a Sigmoidal Function. Mathematics of Control, Signals, and Systems 2, 303–314.
Genesio R. and A.Tesi, Automatica, 28 (1992) 531.
Genesio, R and A.Tesi, Int. J. of Bifurcation and Chaos, 2 (1992) 61.
Glass L. and M.L.Mackey, From Clocks to Chaos. The Rithms of Life, Princeton Un. Press, Princeton, 1988.
Gleick, J. (1987). Chaos: Making the new science. Viking Penguin.
Greenwood, J.A., J.M.Landwehr, N.C.Matalas, J.R.Wallis (1979). Probability weighted moments: Definition and relation to parameters of several distributions expressable in inverse form. Water Resources Research, 15, 1049–1054.
Guyon, I. Boser, B., and V.Vapnik (1992). Automatic capacity tuning of very large VC-dimensions classifiers. Advances in Neural Information Processing Systems V5, pp. 147–155.
Denker, J.S., Le Cun, Y. and S.A.Solla (1987). Optimal Brain Damage. In D.S.Touretzky, editor, Advances in Neural Information Processing Systems asdfasdf, number 2, pages 598–605, San Meteo, Califonia, Morgan Kauffmann.
Dempster, A.P.Laird, N.M. and Rubin, D.B. (1977). Maximum likelihood from incomplete data via EM algorithm, J. Roy. Stat. Soc. B 39:1–38.
Devroye, L. (1985). Expected time analysis of algorithms in computational geometry. In Computational Geometry, Editor, G.T.Toussaint, North-Holland, pages 135–151.
Dibike, Y. (2002). Model induction from data: Towards the next generation of computational engines in hydraulics and hydrology. Swets & Zeitlinger, Lisse.
Dibike, Y.B, S.Velickov, Solomatine D.P. and M.B.Abbot (2001). Model Induction with Support Vector Machines: Introduction and Applications. ASCE Journal of Computing in Civil Engineering, Vol 15, No.3.
Dontchev A., and M.Zollezi (1992). Second order discrete approximations to linear differential inclusions. SIAM J. Numer. Anal., 29(2):439–451.
Drazin, P.G. (1992). Nonlinear Systems, Cambridge University Press.
Fang, S.-C. and Puthenpura, S. (1993). Linear Optimization and Extensions: Theory and Algorithms. Englewood Cliffs, NJ: Prentice-Hall, p. 60.
Feigenbaum, M.J. (1982). Low-dimensional dynamics and the period doubling scenario, in Dynamical systems and chaos, 131–148.
Feigenbaum, M.J. (1983). Universal behavior in nonlinear systems, in Order in chaos. Phys. D 7 (1–3), 16–39.
Fisher, R.A. (1922). On the Mathematical Foun-dations of theoretical Statistics. Phil. Trans. A., 309–368.
Fisher, R.A. (1952). Contributions to Mathematical Statistics. J. Wiley, New York.
Fraeser, A.M. and Dimitradis, A. (1994). Forecasting probability densities by using hidden Markov models, in Time Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley, Reading, MA, pp. 265–282.
Fridman J.H. and Stuetzle W. (1981). Projection pursuit regression. J. Am. Statist Assoc, 76:817–823.
Friedman, J. H (1991). Multivariate Adaptive Regression Splines. Annals of Statistics 19, pp.1–15.
Frison, T.W., Abrabanel, H.D. and M.Earle (1999). Chaos and predictability in ocean water levels. Journal of geophysical research, 104(4) pp. 7935–1951.
Glass, L. and Mackey M.L (1988). From Clocks to Chaos: The Rithms of Life, Princeton Un. Press, Princeton.
Glivenko V.I. (1933). Sulla determinaziione empirica di probabilita. Giornale dell' Instituto Italiano degli Attuari (4).
Ghahramani, Z. and Hinton, G.E. (1998). Switching state-space models, Technical Report CRG-TR-96–3, Department of Computer Science, University of Toronto.

Grassberger, P. (1986). Towards a quantitative theory of self-generated complexity. Int. J. Theor. Phys., 25: 907.
Hadamard, J. (1898). Les surfaces a curbures opposees et tours lignes geodesiques. Journal of Mathematical Analyses and Applications 4:27–73 (reprinted in Oeuvres de Jacques Hadamard. 1968. Paris: Centre National de la Recherche Scientifique).
Hald, A. (1952). Statistical Theory with Engineering Applications. John Wiley, New York.
Haken, H. (1983). Advanced synergetics: Instability Hierarchies of Self-Organizing Systems and Devices Springer.
Hassibi, B. and Stork, D.G. (1993). Second order derivatives for network pruning: Optimal brain surgeon. In Lippman, D.S., Moody, J.E., and Touretzky, D.S., editors, Advances in Neural Information Processing Systems 5, pp. 164–171. Morgan Kaufmann.
Haykin, S. (1999). Neural Networks: A Comprehensive Foundation. New Jersey: Prentice Hall.
Heckerman, D. (1997). Bayesian Networks for Data mining, Data mining and Knowledge discovery 1, 79–119.
Heckerman, D.(1995b). A Tutorial on learning Bayesian Networks, Technical report, Microsoft Research MSR-TR-95–06.
Heckerman, D., Geiger, D.and Chickering, D. (1995a). Learning Bayesian Networks: The combination of knowledge and Statistical, Technical report, Microsoft Research MSR-TR-94–09.
Hermans, L.M., Giampiero E.G.Beroggi and D.P.Loucks (2002). Managing water quality in a New York City watershed. Journal of Hydroinformatics 5, pp. 155–168.
Hetch-Nielsen, R. (1090). Neurocomputing. Reading, MA: Addison-Wesley.
Herzel, H., Schmitt, A.O. and W.Ebeling (1994). Phys. Rev. E 50:5061.
Hornik, W. (1989). Multilayer Feedforward Networks are Universal Approximators. Neural Networks, vol. 2 pp. 359–366.
Holtton, J.R. (1992). An introduction to dynamic meteorology. Third edition San Diego, Cal., Academic Press.
Hosking, J.R.M. (1990). L-moments: Analysis and estimation of distributions using linear combinations of order statistics. J. R. Statist. Soc., Ser. B, 52(1), 105–124.
Hsu, K. Gupta, H.V. and Sorooshian, S. (1995). Artificial Neural Network Modeling of the Rainfall-Runoff Process. Water Resources Research, Vol 31(10), 2517–2530.
Hsu, K. Gupta, H.V. and Sorooshian, S. (1998). Streamflow forecasting using artificial neural networks, Water Resources Engineering 98, Proceedings ASCE Conference, Memphis, Tennessee.
Hsu, S.Masters, T.Kuhl, F.P.Tenorio, M.F.Reeves, A. and Grogan, T. (1991). Comparative analysis of five neural network models. Technical Papers of the ACSM, Vol 5, pp 182–191.
Huber, P.J. (1985). Projection pursuit. Ann. Statist, 13, 435–525.
Huber, P.J. (1964). Robust estimation of a location parameter. Ann.Math. Statist.
Ivanov V.V. (1962). On inear problesms which are not well-posed. Soviet Math. Docl. 3(4), pp. 981–983.
Jacbs, R.A., Jordan, M.I., Nowlan, S.J. and Hinton, G.E. (1991). Adaptive mixture of local experts, Neural Computation 6:181–214.
Jang, J.-S.R., “Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm,” Proc. of the Ninth National Conf. on Artificial Intelligence (AAAI-91), pp. 762–767, July 1991.
Jang, J.-S.R., “ANFIS: Adaptive-Network-based Fuzzy Inference Systems,” IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp. 665–685, May 1993.
Jayawardena, A.W. and Lai, F. (1994). Analysis and prediction of chaos in rainfall and stream flow time series, journal of Hydrology, 153:23–52.
Jensen F. (1996). An introduction to Bayesian Networks, UCL Press Limited, London.
Jensen, F., Lauritzen, S. and Olesen, K. (1990). Bayesian updating in recursive graphical models by local computaions, Computational Statistics Quarterly 4:269–282.

Joachims, T. (1997). Svm light: Implementation of the decomposition training algorithm. Bell Lab. Lucent Technologies.
Jonoski, A. (2002). Hydroinformatics as Sociotechnology: Promoting Individual Stakeholder Participation by Using Network Distributed Decision Support Systems. PhD thesis. Balkema, Rotterdam.
Juditsky, A. (1997). Wavelet estimators: adapting to unknown smoothness, Mathematical Methods of Statistics 6:1–25.
Kehagias A. and Petridis V. (1997). Time series segmentation using predictive modular neural network, Neural Computations, Vol 9, pp. 1691–1710.
Koch, S.P. (1991). Bias error in maximum likelihood estimation. J. Hydrol., 122, 289–300.
Kolmogorov, A.N. (1959). Dokl. Acad. Naul USSR, 124:754.
Krogh, J.A. and Hertz. A. (1992). Simple Weight Decay Can Improve Generalization. Advances in Neural Information Processing Systems, 4, J.E.Moody, S.J.Hanson and R.P.Lippmann, eds., Morgan Kauffmann Publishers, San Mateo CA, pp. 950–957.
Laplace, P.S. Philosophical Essays on Probabilities. Springer-Verlag, New York, 1995. Translated by A.I. Dale from the 5th French edition of 1825.
Laskey, K.B. (1995). Sensitivity Analysis for probability assessments in Bayesian networks, IEEE Transactions on Sytems, Man and Cybernetics, pp. 901–909.
Lauritzen, S. (1992). Propagation of probabilities, means, and variances in mixed graphical association models, Journal of the American Statistical Assocaition, 87:1098–1108.
Lauritzen, S. and Spiegelhalter, D. (1988). Local computation with probabilities on graphical strucutures and their application to expert systems, J. Royal Statistical Society B, 50:157–224.
LeCun, Y. (1986). Learning Processes in an Asymmetric Threshold Network. In Disordered systems and biological organization, (E.Bienenstock, F.Fogelman-Soulie, and G.Weisbuch, eds.), Les Houches, France, pp. 233–240.
LeCun, Y. (1988). A theoretical framework for Back-Propagation. In Proceedings of the 1988 Connectionist Models Summer School, (D.Touretzky, G.Hinton, and T.Sejnowski, eds.), CMU, Pittsburgh, Pa, pp. 21–28.
LeCun, Y., Boser, Y., Denker, Y.D., Henderson, R.E. Howard, Hubbard, W., and L.D.Jackel. (1989). Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541–551.
Ljung, L., Pflug, G., and H.Walk (1992). Stochastic Approximation and Optimization of Random Systems,Birkhäuser, Berlin, 113 pages ISBN 3-7643-2733-2.
Lissier, E.F. (1998). Markov Models and Hidden Markov Models: A Brief Tutorial, Technical report, International Computer Science Institute, TR-98–041.
Lorenz, N. (1963). Deterministic non-periodic flow, J. Atmos. Sci., Vol. 20, pp. 130.
Loucks, D.P, (2000). Modeling the biophysical and social dynamics of a ‘River of Grass’: a challenge for hydroinformatics. Journal of Hydroinformatics 2, pp. 207–217.
Lowe, D. (1989). Adaptive radial basis functionnonlinearities, and the problem of generalization. In 1st IEE International Conference on Artificial NeuralNetworks, pages 171–175, London, UK.
Lu, Z.Q. and Berliner, L.M. (1999). Markov switching time series models with application to a daily runoff series, Water Resources Research, 35(2):523–534.
Lundgren J.T. (1987). An algorithm for the combined distribution and assignment model. Report, Department of Mathematics, Linkoping University, Linkoping, Sweden.
Mathews, V.J. (1991). Adaptive polynomial filters. IEEE Signal Processing Magazine, vol. 8, no. 3, pp. 10–26.
Minns, A.W. (1995). Analysis of experimental data using artificial neural networks. HYDRA 2000, Proc. XXVI Congress IAHR, London, Vol. 1. Thomas Thelford, London, pp. 218–223.
Minns, A.W and Hall, M.J (1996). Artificial neural networks as rainfall runoff models, Hydrological Science Journal, vol. 41(3).
Minns, A.W. (1998). Artificial Neural Networks as Subsymbolic Process Descriptors. A.A Balkema, Rotterdam, The Netherlands.

Moody, J., and C..J. Darken (1988). Learning with localized receptive fields, In Proceedings of the 1988 Connectionist Models Summer School. pp. 133–143, editors: Touretzky et.al., Morgan-Kaufman.

Mitchell, T.M. (1997). Machine Learning, The McGraw-Hill Companies, Inc.

Muller, K.R., Smola, A.J. , Ratsch, G., Scholkopf, B.,. Kohlmorgen, J. and V.Vapnik (1997). Predicting Time Series with Support Vector Machines. Proceedings of ICANN '97.

Mynett, A.E. (2002). Environmental Hydroinformatics: the way ahead, Proceedings of the Fifth International Conference on Hydroinformatics, Cardiff, UK, 2002, Vol. 1, pp 31–36, IWA, London.

Mynett, A.E. (2001). Hydroinformatics in Aquatic Resource Management, invited presentation at the NATO Advanced Research Workshop on New Paradigms in River and Estuary Management, Boise, Idaho, USA.

Mynett, A.E.; Chen, Q.; Blauw, A.N. (2002). Fuzzy logic and artificial neutral network modelling of phaeocystis in the north Sea, Proceedings of the Fifth International Conference on Hydroinformatics, Cardiff, UK, 2002, Vol. 1, pp. 722 Ð728.

Mynett A.E., Price R. (2000). Hydroinformatics and knowledge management,, Hydroinformatics 2000, cedar Rapids, IA, USA.

Neapolitan, R.E. (1990). Probabilistic Reasoninig in Expert Systems, Theory and Algorithms, A Wiley-Interscience publication, U.S.A.

Nelder, J.A and R.W.M.Wedderburn (1972). Generalized linear models. Journal of the Royal Statistical Society A 135:370–384.

Nicolis, C.,. Ebeling, W. and C.Baraldi (1997). Tellus 49A: 10.

Osuna, E., Freund, R., Girosi, F. (1997). Training Support Vector Machines:An Application to Face Detection. Proceedings of CVPR'97.

Pao, Y.H. (1989). Adaptive Pattern Recognition and Neural Networks. Addison-Wesley, Reading, MA.

Parzen, E. (1962). On estimation of a probability density function and mode. Annals Mathematical Statistics Vol. 33, pp. 1065–1076.

Pawelzik, K., Kohlmorgen, J. and Müller, K.R. (1996). Annealed competition of experts for a segmentation and classification of switching dynamics, Neural Computation, Vol 8. pp. 340–356.

Phillips, D.Z. (1962). A technique for numerical solution of certain integral equations of the first kind. J. Assoc. Comput. Mach., 9, pp. 84–96.

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, CA.

Plaut, D.C., Nowlan, S.J. and Hinton, S.J. (1986). Experiments on learning by back-propagation. Technical report CMU-CS-86–126, Carnegie-Mellon University, Pittsburgh, PA.

Poncaré, H. (1908). Science and Method. Translated by Francis Maitland, Dover.

Price, R.K. (2000). Hydroinformatics and urban drainage: an agenda for the beginning of the 21st century. Journal of Hydroinformatics, Vol. 2, No.2, pp. 133–147.

Price, R.K. (2001). Hydroinformatics, modelling and knowledge management. White paper, IHE-Delft.

Poggio, T. and Girosi, F. (1990). Network for approximation and learning. Proceedings of the IEEE, 78:1481–1497.

Rabiner, L.R. and Juang, B.H. (1986). An introduction to hidden Markov models, IEEE Acoustics, Speech & Signal processing Magazine, 3:4–16.

Rabiner, L.R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77:257–286.

Refenes, A. (1995). Neural Networks in the Capital Markets. JohnWiley and Sons, 1995.

Refsgaard, J.C., Knudsen, J. (1996). Operational validation and intercomparison of different types of hydrological models. Water Resources Research, vol. 32, no. 7, pp. 2189–2202.

Ripley, B.D. (1996). Pattern Recognition and Neural Networks. Cambridge.
Robinson, T. (1994). An application of recurrent nets to phone probability estimation. IEEE Transactions on Neural Networks, 5(2):298–305.
Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. Annals Mathematical Statistics V. 27, pp. 832–837.
Rosenblatt, F. (1962). Principles of Neurodynamics. Washington, DC: Spartan Books.
Ruelle, D. (1979). Thermodynamic Formalism, Addison-Wesley, Reading.
Ruelle, D. and Takens, F. (1971). On the nature of turbulence. Commun. Math. Phys. 20, 167–192. 23, 343–344.
Rumelhart, D.E., Hinton, G.E. and R.J.Williams (1986). Learning internal representations by back-propagating errors. In Parallel Distributed Processing: Explorations in the Microstructure of Cognition, D.E. Rumelhart and J.L.McClelland, Eds. Cambridge, MA: MIT Press, vol. 1, pp. 318–362.
Savenije H.H.G. (1993a). Composition and driving mechanisms of longitudinal tidal average salinity dispersion in estuaries. Journal of Hydrology, 144, pp. 127–141.
Savenije H.H.G. (1993b). Predictive model for salt intrusion in estuaries. Journal of Hydrology, 148, pp. 203–218.
Saunders, C, Stitson, M.O., Weston, J and L.Bottou (1998). Support Vector Machine Reference Manual. Department of Computer Science, Royal Holloway University of London.
Schiner, J.S., Davison, M. and P.T.Landsberg (1999). Phys. Rev. E 59:1459.
Scheffer W.M. and M.Kot, “Differential systems in ecology and epidemiology,” in Chaos (A.V. Holden ed.), Manchester Un. Press, Manchester, 1986, pp. 158–178.
Schmidt, M. (1996). Identifying Speakers with Support Vector Machines. In Proceedings of Interface ’96, Sydney.
Schölkopf, B., Sung, K., Burges, C., Girosi, F., Niyogi, P., Poggio, T., and Vapnik, V. (1997). Comparing support vector machines with Gaussian kernels to radial basis function classifiers. IEEE Trans Sig Proc, 45(11):2758–2765.
Schölkopf, C., Burges, J.C., and Smola, A.J. (1999). Advances in Kernel Methods. MIT Press.
Shannon, C. (1951). Bell Systems Techn. 30:50.
Shrestha, D.L. (2002). Application of Bayesian Networks in Hydroinformatics, M.Sc. Thesis H.H 428, IHE, Delft.
Sinai, Y.B. (1959). Dokl. Acad. Naul USSR, 124:754, 125:1200.
Sivakumar, B. (2000). Chaos theory in hydrology: important issues and interpretations, Journal of Hydrology, 227(1–4), 1–20.
Sivakumar, B.R.Berndtsson, J.Olsson, K.Jinno, A.Kawamura (2000). Dynamics of monthly rainfall-runoff process at the Gota basin: A search for chaos, Hydrology and Earth System Sciences, 4(3) 407–417.
Sivakumar, B., Wallender, W.W., Puente, C.E. (2003). Characterization of monthly streamflow dynamics in the western United States. Proceedings of the American Geophysical Union Fall Meeting, December 8–12, San Francisco, USA.
Smola, A. (1996). Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Technical Report No. 44. Max-Planck-Institut für biologische Kybernetik, Tübingen.
Stive, M.J.F. and A.J.H.M.Reniers (2003). Sandbars in motion, Science, 299, 1855–1856.
Stive, M.J.F., Wang, Z.B., Capobianco, M., Ruol, P. and Buijsman, M.C. (1998). Morphodynamics of a tidal lagoon and the adjacent coast, in: Physics of Estuaries and Coastal Seas, Dronkers & Scheffers (eds), Balkema, Rotterdam, pp 397–407.
Stive, M.J.F. and De Vriend, H.J. (1995). Modelling shoreface profile evolution, Marine Geology, 126, 235–248.
Stutz, J. and Cheeseman, P. (1994). AutoClass—a Bayesian Approach to Classification. In Maximum Entropy and Bayesian Methods, Cambridge 1994, Kluwer Academic Publisher.

Solomatine, D.P and Avila Torres, L.A, (1996). Neural Network Approximation of a hydrodynamic Model in Optimizing reservoir operation, Proceedings of Hydroinformatics '96 conference, pp 201–206.
Solomatine D.P. Two strategies of adaptive cluster covering with descent and their comparison to other algorithms. Journal of Global Optimization, 1999, vol. 14, No. 1, pp. 55–78.
Solomatine, D.P., Velickov, S. and J.C.Wüst (2001). Predicting water levels and currents in the North Sea using chaos theory and neural networks. Proc. 29th IAHR Congress, Beijing, China.
Solomatine, D.P. and K.N.Dulal (2003). Model trees as an alternative to neural networks in rainfall-runoff modeling Hydrological Sciences Journal, 48(3), pp. 399–411.
Solomatine, D.P. and Y.Xue (2004). M5 model trees compared to neural networks: application to flood forecasting in the upper reach of the Huai River in China. ASCE Journal of Hydrological Engineering (submitted).
Tikhonov, A.N. (1963). Solution of incorrectly formulated problems and the regularization method. Soviet Math. Dokl., 4:1035–1038.
Tsonis, A.A. (1992). Chaos from theory to applications, Plenum Press, New York.
Tsonis A.A. and J.B.Elsner (1988). The Weather Attractor Over Very Short Time Scales, Nature 33:545.
Vapnik, V. and A.J.Chervonenkis (1968). On the uniform convergence of relative frequencies of events to their probabilities.. Doklady Akademii Nauk USSR 181(4). (English translation).
Vapnik, V. and A.J.Chervonenkis (1971). On the uniform convergence of relative frequencies of events to their probabilities. Theory Probab. Appl. 16 pp. 264–280.
Vapnik, V. and A.J.Chervonenkis (1989). The necessary and sufficient conditions for consistency of the method of empirical risk minimisation. Pattern Recogn. On Image Analysis 1(3) pp. 284–305.
Vapnik, V. and A.R.Stefanyuk (1978). Nonparametric methods for restoring probability densities. Avtomatika i Telemekhanika, (8):38–52.
Vapnik, V. (1979). Estimation of dependences based on empirical data. Nauka, Moskow.
Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer-Verlag, New York.
Vapnik, V. (1998). Statistical Learning Theory. Wiley.
Vassilios, P. and A.Kehagias (1998). Predictive modular neural networks : applications to time series. Boston: Kluwer Academic Publishers.
Vriend de, H.J. and Stive, M.J.F. (1988). Quasi-3D modelling of nearshore currents. Coastal Engineering, 11: 565–602
Velickov S. (1997). Distributed hydrological modelling using GIS. Macedonian water management journal, No.13, Skopje.
Velickov S. (1998). Terminology, modelling protocol and classification of the hydrological models. Macedonian water management journal, No.14, Skopje.
Velickov, S., D.P.Solomatine, R.K.Price and Yu X.(2000). Application of Data Mining Technologies for Remote Sensing Image Analysis. Hydroinformatics '2000–4th International conference on hydroinformatics, Iowa City, USA.
Velickov, S. and D.P.Solomatine (2000). Predictive Data Mining: Practical examples. 2nd Workshop on Application of AI in Civil Engineering. Cottbus, Germany.
Velickov, S. and Solomatine D.P. (2002). Nonlinear dynamics, deterministic chaos and predictability in the North Sea water levels along the Dutch coast. Technical report DC-07.04.02.
Velickov, S. (2003a). Predicting water levels in the North Sea using theory of nonlinear dynamics and chaos. IASTED International Conference on Modelling and Simulation, Palm Springs, USA.
Velickov, S. (2003b). Mixture of Models: A New Framework For Modelling Complex Nonlinear Dynamical Systems, XXX IAHR Conference, Greece.

Velickov, S. (2003c). Data-Driven Modelling Of Nonlinear Dynamical Systems Using Mixture Of Models Framework. The 16th International Conference on Industrial and Engineering Applications of Artificial Intelligence & Expert Systems, Loughborough, UK.

Velickov, S, Price, R and D.Solomatine. (2003). Prediction of Nonlinear Dynamical Systems Based on Time Series Analysis: Issues of Entropy, Complexity and Predictability. XXX IAHR Congress, Thessaloniki, Greece.

Velickov S. (2004a). Chaos in Rainfall? Taking Time Seriously. Journal of Hydrologic Engineering (accepted, to appear).

Velickov S. (2004b). Chaos and predictability in the North Sea water levels along the Dutch cost. Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere (accepted, to appear).

Weigend, A.S., Mangeas, M and A.N.Srivastava (1995). Non-linear gated experts for time series, International Journal of Neural Systems, vol.6, pp. 373–399.

Werner, P.C., Jimenez-Montano, M.A. and T.Pohl (1999) Changes in Probability of Sequences, Exit Time Distribution and Dynamical Entropies in Potsdam Temperature Records, Theor. Appl. Climatol. 62:125.

WMO (1975). Inter-comparison of conceptual models used in operational hydrological forecasting, Technical Report No 429, Geneva, Switzerland.

Wolfgong, M. (1999). Stochastically Based Semantic Analysis, Kluver Academic Press, Boston.

Wüst, J.C, (1995). Current prediction for Shipping Guidance, Neural Networks: Artificial Intelligence and Industrial Applications, Kappen, B and Gielen, S (eds), Springer, London.

Zijderveld, A. (2003). Neural Networks Design Strategies and Modelling in Hydroinformatics. PhD thesis, TU Delft, The Netherlands.

Zhang, Q. (1993). Regressor selection and wavelet network construction. In IEEE Conference on Decision and Control (CDC), San Antonio, USA.

# About the author

Slavco Velickov received his B.Sc. degree in Hydraulic Engineering from “St. Ciril & Methodius” University at the Faculty of Civil Engineering in Skopje, Macedonia in 1993. From 1993 till 1998 he worked as an assistant professor at the Faculty of Civil Engineering teaching computational hydraulics, hydrological modeling and river engineering. During the period of 1994–1996 he finished postgraduate studies at the same University and successfully defended his M.Sc. thesis entitled as “Distributed Hydrological Modelling using GIS”. In that period he carried out more than 25 research and general design engineering projects. As from September 1996 he joined the Hydroinformatics section at the International Institute for Infrastructural, Hydraulic and Environmental Engineering (IHE) and also obtained M.Sc. degree in hydroinformatics in 1998 related to internet computing.

Starting from June 1998, he continued to work at IHE (currently UNESCO-IHE Institute for Water Education) as a lecturer in hydroinformatics. In addition to teaching activities, he was also involved in various research projects including supervision and guidance of students. The main direction of his PhD research included development and applications of new emerging modelling techniques based on the theory of nonlinear dynamics and deterministic chaos to water and environmental systems. His broader research topics included: artificial intelligence, machine learning, chaos theory, wavelet analysis, artificial neural networks, fuzzy-logic, support vector machines, multi-agent and knowledge-based systems, internet computing, and knowledge engineering and management.

As from September 2003 he joined the SKF Research and Engineering Centre in Nieuwegein. He is currently responsible for the research, development and implementation of knowledge-based systems for various industrial applications, ranging from industrial hydraulics, mechatronics and new materials development to automotive industry. He has published more than 20 scientific papers in international journals and conference proceedings and has contributed as a reviewer in several international journals.