2000

PROGNOSIS - Historical Pattern Matching for Economic Forecasting and Trading

BANAVAS, GEORGIOS NIKOLAOS

http://hdl.handle.net/10026.1/1642

http://dx.doi.org/10.24382/1356
University of Plymouth

PROGNOSIS - Historical Pattern Matching for Economic Forecasting and Trading

by

GEORGIOS NIKOLAOS BANAVAS

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing
Faculty of Technology

August 2000

1st Supervisor: Dr. Susan Denham

2nd Supervisor: Prof. Dr. Michael Denham

3rd Supervisor: Dr. Angelo Cangelosi

Date of Submission: 30. August 2000

To my parents,

Maria and Nikolaos Banavas

Author's declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

This study was financed by the Higher Education Funding Council of England (HEFCE Qr).

A programme of advanced study was undertaken, including a postgraduate course in financial modelling (MSc Computational Intelligence - COIN511).

Relevant scientific seminars and conferences were regularly attended at which work was often presented; external institutions were visited for consultation purposes and several papers prepared for publication.

Publications

- G. N. Banavas, S. Denham and M. J. Denham (2000), "Fast non-linear deterministic forecasting of segmented stock indices using pattern matching and embedding techniques.", submitted to CEF2000, Barcelona.

- G. N. Banavas, S. Denham and M. J. Denham (2000), "A Financial Stock Index Prediction approach with temporal considerations.", invited for presentation to the joint CF2000/FFM2000 conference, London.

- G. N. Banavas et al (2000), "Intelligent monitoring and communication system", Proceedings of SPIE Vol. 3995: Nondestructive Evaluation of Highways, Utilities, and Pipelines IV, Newport Beach, California.

- G. N. Banavas (1999), "Robust Direction Movement Forecasting using a Pattern Matching method applied on Univariate Financial Time Series.", ISF99, Washington DC.

Presentations and Conferences attended

- 6th International Conference on Computing in Economics and Finance (CEF2000) - Society for Computational Economics, Barcelona 2000.
- 7th International Joint Conference on Computational Finance and Forecasting Financial Markets (CF/FFM2000), London, 2000
- 19th International Symposium of Forecasting (ISF99), Washington DC, 1999
- Centre for Neural and Adaptive Systems (CNAS) Industrial Applications Seminars series 1998-99, 1999-2000

Signed .....

Date 03.10.2000

Abstract

In recent years financial markets have become complex environments that continuously change and they change quickly. The strong link between the continuous change in the markets and the danger of losing money when trading in them, has made financial studies a domain that concentrates increasing scientific and business attention. In this context, the development of computational techniques that can monitor recent financial events can process them according to their similarity with historical data recordings, and can support financial decision making, is a challenging problem.

In this work, the principal idea for tackling this problem is the integration of 'current' market information as derived from the market's recent past and historical information. A robust technique which is based on flexible pattern matching, segmented data representations, time warping, and time series embedding dimension measures is proposed. Complementary time series derived features, concerning trend structures, temporal considerations and statistical measures are systematically combined in this technique. All these components have been integrated into a software package, which I called PROGNOSIS, that can selectively monitor its application and allows systematic evaluation in terms of financial forecasting and trading performance.

In addition, two other topics are discussed in this thesis. Firstly, in chapter 3, a neural network, that is known as the *Growing Neural Gas* network, is employed for financial forecasting and trading. To my knowledge, this network has never been applied before to financial problems. Based on this a neural network forecasting and trading benchmark

was constructed for comparison purposes.

Secondly, a novel method of approaching the well established *co-integraton* theory is proposed in the last chapter of the thesis. This method enhances the co-integration theory by integrating into it local time relations between two time series. These local time dependencies are identified using dynamic time warping. The hypothesis that is tested is that local time shifts, delays, shrinks or stretches, if identified, may help to reveal co-integrating movement between the two time series. I called this type of co-integration *time-warped co-integration.* To this end, the time-warped co-integration framework is presented as an error correction model and it is tested on arbitrage trading opportunities within PROGNOSIS.

Preface

Since the early nineties, advanced computer technologies have continuously worked their way into financial service organizations in order to establish and increase their competitiveness in the global economy. For the last few years they have also gained ground in the domain of personal finance, due to the broad expansion of computer technology and telecommunication services. The challenging environment of finance, the high rate of its potential expansion and the vast amounts of money that are invested in the development of advanced computer technology for financial applications, have attracted many researchers from different disciplines. Financial analysts, economists, biologists, psychologists, engineers and computer scientists form a community whose aim is to study the markets, to explore their underlying structure and to develop computer based systems that can support investment decisions and generate profitable trading. The interdisciplinary work presented here represents an attempt to add a small contribution to this quest.

I was granted the opportunity to work in the group of Prof. Denham at the *Centre for Neural and Adaptive Systems* of the School of Computing in the University of Plymouth. Thanks to all the members of this group for the superb working conditions and the creative atmosphere. I am deeply indebted to my advisor, Susan Denham for supervising my work and for providing an implicit assurance that I could investigate new computational finance ideas without fearing I find myself alone. I am also grateful to Michael Denham, Angelo Cangelosi, Guido Bugmann and Roman Borisyuk for investing part of their time to answer the questions I posed to them and for invaluable comments on earlier drafts.

Linda Lanyon is thanked for her 'correct English' language suggestions, since my native language is not English and I do not claim that I command the language. Needless to say that any mistakes that remain in the text are my own.

I would also like to convey my appreciation to Bernd Fritzke, Christoph von der Malsburg and the '97 class of the *Institut für Neuroinformatik*, in Ruhr Universität, Bochum, Germany for guiding my first research efforts.
Of course, I finally wish to acknowledge the financial support of several organisations - the University of Plymouth, the British Royal Academy of Engineers and the European Research Council.

This thesis was typeset in LaTeX. I would like to thank the authors for this convenient and free software tool. "*Thank you all*"!

Georgios N. Banavas

Plymouth, August 2000

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Although financial markets are seemingly based on simple rules of demand and supply, the procedure of making forecasting based investment decisions for trading is a complicated task. A set of major scientific domains, such as sociology, psychology, theoretical economics, financial engineering and computer science, contibute to solving this problem from different angles. This pluralism of approaches together with the potential profit that can be gained from a successful investment decision, make the development of a forecasting based decision support system an interesting and challenging problem. But what makes it so difficult to build robust, in terms of profit, automated forecasting decision support systems (AFDSS) for investment and trading? Let me emphasise some major points.

Firstly, in complex financial environments, it is very difficult to forecast and therefore to take trading positions, by trying to anticipate a decision making process, which will begin and be taken by another group of people at unknown time. Even if all information, related to an investment decision was publicly available, there is no way for somebody to know how another group of people will interpret that information. In that respect valuable information, or technically speaking, necessary input for an AFDSS is missing. Secondly, an AFDSS must operate within markets, which, up to a certain point, are believed to be efficient (MALKIEL, (1996), [111]) and continuously change. For these reasons to perform forecasts necessary for investing in the market is hard work. It is interest-

ing at this point to quote Burton Malkiel's [111] words: "*Investing is a gamble whose success depends on the ability to predict the future*". E. PETERS [134] [135], in an effort to explain the underlying change in the dynamics of the market, quoted George Carlin's words: "*The present does not exist because as soon as we are aware of the present, it is already the past, and the future is the present.*"

Thirdly, huge amounts of collected data make efficient data processing and computational time reduction a task that needs to be addressed. Qualitative data selection techniques and local data processing ideas that are based on long memory statistics, allow us to process considerable data collections in "finite" time.

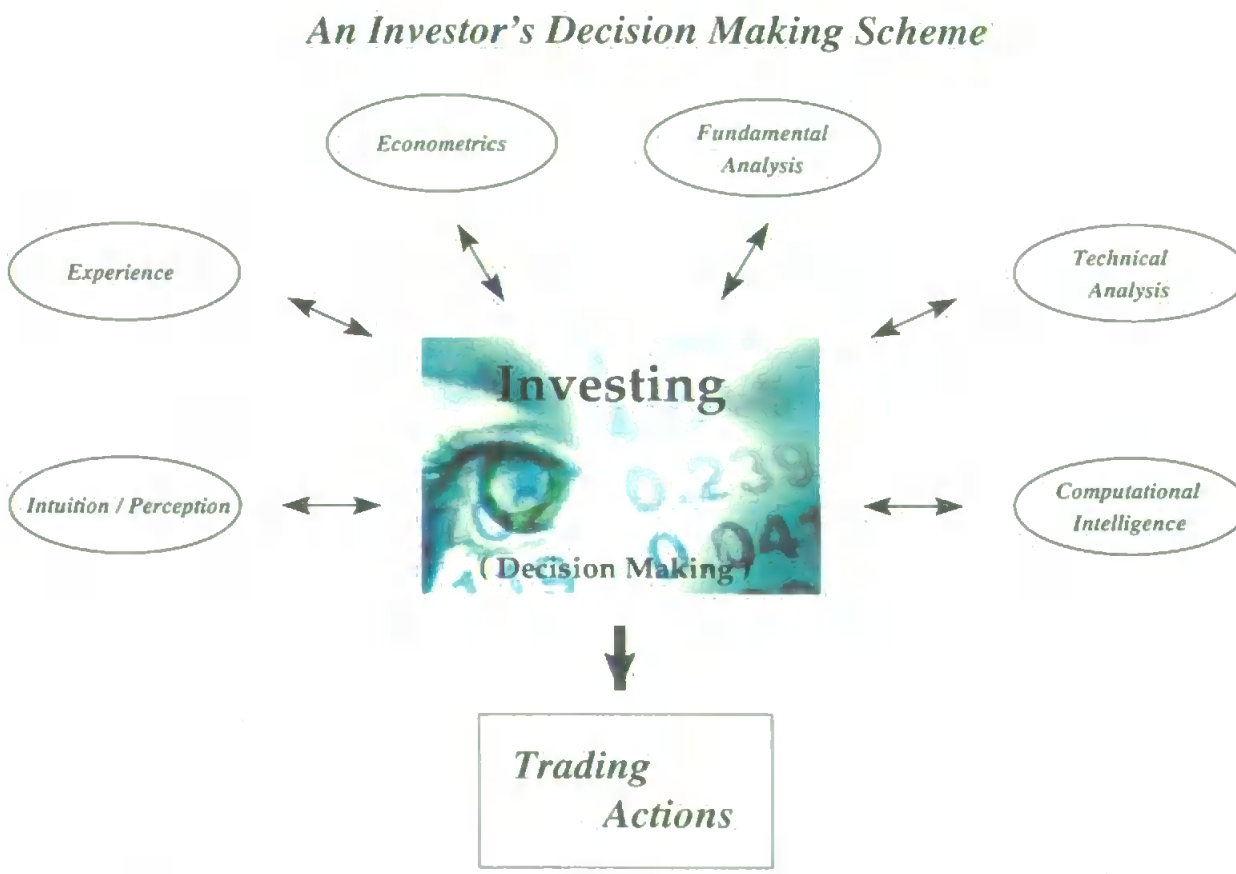


Figure 1.1: An investor integrates many strategies in order to make capital investment decisions and take trade actions.

In my opinion, professional investors employ six main strategies for managing their

capital, illustrated in figure 1.1:

- **Fundamental Analysis:** A fundamentalist investor sharply distinguishes a stock's current price from its real value by mainly interpreting the corresponding firm's annual report. S/He analyses price/earnings ratios, dividends, sales levels, operating costs and other account indicators, to produce valuable insights which might determine the stock's future value which is not reflected in its current price. However, the sole use of fundamental analysis could prove dangerous for investing due to inaccurate data (consider accounting manipulations of annual reports), faulty interpretations, careless timing and unexpected market behaviour (Markets do not always converge to the fundamental analysis based estimates).

- **Econometrics:** Econometrics is the name given to the study of quantitative tools for analysing economic data. The field of econometrics is based on probability and statistical theory. Its ultimate goal is to identify and measure relationships between economic variables, to test the theoretical ideas behind these relationships, and to use them for quantitative predictions. To a greater or lesser extent, econometrics seeks to provide answers to questions that fundamental analysis leaves unanswered. However, it can not be used independently for investing and trading. Its increasing mathematical complexity does not offer solutions that can be easily interpreted and therefore is difficult apply.

- **Technical Analysis:** A technical analyst (chartist) invests capital while assuming that all information about earnings, dividends and stock future performance of a company is automatically captured in the chart that depicts past prices and trading volumes. Additionally, he/she believes that prices tend to move in trends whose direction of change can be foreseen and can lead to successful trading positions ([159], [142], [143], [121]). The main arguments against the sole use of Technical Analysis (TA) can be summarised in the following:

  TA indicators tend to follow the time series trend rather than predict it. That makes them vulnerable to sharp changes. Due to its simplicity, TA is gradually

used by more and more people. This depreciates its validity. Finally, due to the efficiency of the market, a price chart's reading can be quickly anticipated and this reduces the chartist investor's opportunity to act profitably.

- **Computational Intelligence:** The strength of computational intelligence and machine learning is that it can be applied to domains where little is known about underlying relationships. In finance, applied computational intelligence gives to an investor a genuine replacement for the non-adaptive "*a priori*" statistical forecasting models of econometrics and the structural models of technical analysis. However, financial modelling using learning techniques seems to be one of the hardest tasks of learning systems technology and usually must be accompanied with valuable "*a priori*" knowledge derived from experts, in order to overcome the two main problems in machine learning applications: convergence to local minima and architecture (REFENES (1993), [145], pp. 778).

- **Experience:** Investors also use their experience to judge market behaviour and manage their capital. Past profit and loss situations that they have experienced may be considered as helpful indications for participating in the market by going *long* (buy) or *short* (sell), or staying out of it. However, experience on its own cannot be considered as a successful investment strategy because it cannot necessarily be applied to new market conditions. Furthermore, experience based trading heavily depends on human memory.

- **Intuition/Perception:** Investors and many of the ordinary market players base their positions on intuition and perception. They simply "feel" that a company's share will rise. They usually base these intuitive opinions on a political decision, on publicly available news and rumors, or on the way that a company and its shares are seen through their eyes. Over-trading, premature liquidation of good positions, "jumping the gun" on market entry to get a better price, holding in a losing position are all negative manifestations of intuition and subjectivism in actual trading.

The emphasis of this work will be on the computational intelligence component of a forecasting based investment decision support procedure, although the other factors will also be touched upon.

More precisely, we will employ pattern recognition and matching technology, inspired by the investor's experience and perceptions, in an effort to identify market situations which have occurred in the past and which are similar or related to the present. Our aim is to keep our research as close as possible to the realistic way that a professional investor acts in the market. One of the advantages of this approach, which will be discussed later in this thesis, is that it eliminates manifestations of intuition and subjectivism and helps the investor to avoid many trading errors. Furthermore, an AFDSS removes the need for constant investment decision making and, thus, substantially reduces trading related stress and anxiety.

To test our thoughts, we have chosen to evaluate our algorithms on realistic forecasting and trading tasks. Data for that purpose has been collected from the DATASTREAM Research Services database [44] and all algorithms have been developed in object oriented C++ code on a Pentium Pro-200 machine running under RedHat 6.1 Linux. The data set that has been used for the simulations in this thesis consists of some European market indices as have been calculated by the DATASTREAM team.

The thesis is organized as follows:

We begin by discussing the problem of developing automated intelligent decision support systems based on pattern recognition (chapter 2). The literature in the field is reviewed and fundamental problems for forecasting and trading are identified. We then apply an artificial neural network, the **Growing Neural Gas (GNG)**, to the problem of forecasting financial market indices. The forecasts are also evaluated within a simple trading strategy (chapter 3). In this context, we construct a forecasting and trading benchmark, compatible with other applications of artificial intelligence in financial engineering. The GNG network has never been applied to time series forecasting problems before. In this context, our contribution in chapter 3, is the assessment of the GNG networks as financial predictors and auto-traders. The following chapters describe a pattern match-

ing algorithm for forecasting and trading which operates firstly (chapter 4) on raw data representations based on similarity search mechanisms that use compound feature sets. Secondly, (chapter 5) pattern matching for forecasting and trading is applied on piecewise linear segment time series representations using **Multiple Feature Sets (MFS)** and **Dynamic Time Warping (DTW)**. The use of piecewise linear segment representations for extraction of time dependencies in financial time series; the introduction of the **Embedding Segment Dimension (ESD)** calculation that indicates non-linear deterministic behaviour in segmented financial time series; the introduction of multiple feature sets for univariate time series pattern matching using Dynamic Time Warping; and the use of historical pattern activity of the market for trend prediction and trading, are considered our major contributions that are given in chapters 4 and 5. Next, we study the forecasting and investment decision making problem on the multivariate case of financial data, by again employing pattern matching ideas on segmented data (chapter 6). In the same chapter, we address the notion of co-integration between economic time series and we present the concept of time-warped co-integration which is based on local time relations between the time series. Statistical arbitrage is also discussed there (section 6.2). Our major contribution in this section is included in our effort to explain that local time relations, like local delays or shifts, exist in economic time series. We identify such relationships using the Dynamic Time Warping (DTW) algorithm and we show how co-integration between economic time series can be revealed out of the time warped series representation. We call this type of co-integration **time-warped co-integration**. To identify profitable trading opportunities within the *time-warped co-integration* economic time series movement, we apply an arbitrage derived trading strategy, called statistical arbitrage. Improvement in terms of average profit is shown for *time-warped co-integration* compared to classic co-integration.

All the approaches mentioned above have been integrated within a self developed software package, which I called PROGNOSIS[1]. Information about the PROGNOSIS software

[1] *Prognosis* (*πρόγνωσις*): Greek word which means knowledge, understanding of the future before it arrives - *pro* (*πρό*): before, beforehand, *gnōsis* (*γνῶσις*): knowledge, understanding, [74].

package together with snapshots are given in appendix D.

Chapter 2

Pattern Recognition in Finance

"It is said that the present is pregnant with the future"

Voltaire,
'The Portable Voltaire'

2.1 Introduction

Why should one study financial modelling and forecasting with computerized intelligent pattern recognition methods? The most important reason is the need for technology which will provide investors with automated modelling and forecasting systems, upon which to objectively base their capital investment decisions. Very often investment decisions, based solely on subjective forecasts, are false and lead to substantial loss of money. Econometrics, "*the application of mathematical statistics to economic data, to lend empirical support to the models constructed by mathematical economics and to obtain numerical estimates*" (SAMUELSON et al., (1954), [153]), has tried to put empirical flesh and blood on theoretical structures (JOHNSTON, (1984), [85]). It has had, however, difficulties in coming to terms with economic and financial time series analysis because there is no rich recording of theoretical or empirical research extending over the decades and centuries on which to base a modelling approach (DICKENSON, (1974), [50]). Another reason for the employment of computational techniques in finance arises from the fact that the most

reliable domain to apply and test any theoretical ideas is the real financial world, since the difficulties of complex financial and economic environments arise there naturally. The ability to run realistic simulations has been possible only very recently with researchers implementing their ideas using powerful computing machines.

The nature of any financial applications to be studied, requires careful consideration of a range of economic factors, such as interest and inflation rates, political and central bank economic decisions, social investment and trading, general and public investing trend growth, and so forth. The selective attribute of adaptive pattern recognition forecasting systems introduces, therefore, the idea of automatically isolating significant determinants of the financial process. Furthermore, the adaptive nature of these systems also overcomes the strong version of the Efficient Market Hypothesis (EMH), which actually suggests that no "*a priori*" model for predicting market movements exists (KINGDON, (1997), [97]). This success of artificial intelligence against the strong version of EMH, has been a motivation for further research in analysing financial markets. A relatively new conference series, the COMPUTATIONAL FINANCE ([36, 35]), dealing with such applications, is now held annually.

The most popular learning machines used for financial modelling and forecasting are the family of artificial neural networks (ANN) and genetic algorithms (GA). A particular neural network, the Growing Neural Gas (GNG) [60], [63], considered as a pattern recognition device, is tested and evaluated on market index forecasting in chapter 3. We use the resulting framework obtained from the GNG to create a forecasting and trading performance benchmark.

The system we propose in this thesis is an automated pattern recognition system, which is based on similarity measures and flexible matching. This system is designed to perform forecasting and to generate trading position signals. We call this Pattern Matching Forecasting System PROGNOSIS[1]. PROGNOSIS' forecasts are derived from the "current" situation of a financial assets series and its similarity with historical situations. Behavioural psycholgists have found that, when faced with incomplete input informa-

[1]Greek word which actually means "*before knowledge*", i.e. prediction.

tion, people often base their decisions on the similarity with past experience (PETERS, (1999), [134]). PROGNOSIS' aim is to computationally automate that behaviour and still remain close to the market's efficiency. Consider for example that EMH's future indicator is the last time series value, while that of PROGNOSIS is the time series most recent situation. In chapters 4, 5 and 6 we demonstrate the development and evaluation of PROGNOSIS.

2.2 Requirements

Developing and employing computerized systems to forecast financial time series and to extract investment decisions, is not only a challenging task but also a difficult one. The fact that in order to evaluate a financial system, money is needed[2], together with the considerable complexity of the financial application domain, results in the need for such an automated system to obey the following requirements:

1. **Efficient Market Hypothesis (EMH) Constraints:** Although, the idea has become fashionable in academic literature that financial markets may after all display some signs of predictability (LO et al., (1990), [105], BROCK et al, (1992), [22], MEULBROEK, (1992), [115], RIDLEY, (1993), [148], PETERS, (1994), [135]), any computational intelligence system must be tested against the EMH. In terms of trading, the system must be tested against a simple "*buy-and-hold*" strategy. Remarkable studies on the EMH and its validity - pros and cons - can be found in MALKIEL, (1996), [111] and LO and MACKINLAY, (1999), [106].

2. **Data Optimisation:** When studying the computational intelligence technology, one becomes familiar with terms like data smoothing, detrending, filtering etc. (BISHOP, (1996), [17]). In finance, however, too much preprocessing is not desirable.

[2]That corresponds to real trading. The purpose of any trading system is to operate in the present financial context and to be profitable. Insufficiently tuned trading systems which do not meet certain modelling requirements, are not profitable and therefore when applied they lead to extensive losses.

Consider, for example, the two conflicting notions of trend in financial time series: Trend can be considered as a valuable factor of price movements which might be proven profitable and must be retained but it can also be seen as an obstacle in the learning process of a neural network which must be removed. It is crucial, after all, for a forecasting and investment decision making system to process data that is as close as possible to its natural state.

3. **Adaptivity:** Any investor and thus any system which supports an investor's decisions will face an ongoing changing financial environment. It is thus a requirement of any forecasting and trading system to be able to adapt to new market situations (KINGDON, (1997), [97]).

4. **Robustness:** The robustness of a system in terms of performance is strongly linked with its parameter set. A system heavily dependent on its parameters is not stable. That happens when small changes in the parameter set of the system correspond to unpredictable responses. BROCK, LAKONISHOK and LEBARON, (1992), [22], have shown that some popular chart analysis techniques, such the Moving Average (MA) rule (PRING, (1991), [142], MURPHY, (1999), [121]) may generate profitable trading only under certain parameterisation.

5. **Computational Efficiency:** The financial time series analysis must allow a huge amount of data to be processed. This is vital for obtaining results of statistical significance. It is, therefore, essential for a financial forecasting system to put before any other requirement, process efficiency and computational time reduction.

6. **Simplicity:** D. Hendry, during his plenary session in the International Symposium of Forecasting (ISF'99) in Washington DC, while commenting, on the huge expansion, in terms of complexity, of the econometric models, emphasised the necessity of keeping modelling in finance as simple as possible. Complexity is also an argument that economists raise against neural network technology. They complain of not being able to understand the underlying mechanism when applied for financial

modelling. To some extent they are correct (see ZAPRANIS and REFENES, (1999), [185]). It is important to remember that neural network complexity is usually growing against generalization - the *Bias-Variance* dilemma.

7. **Monitoring Constraints:** A drawback of neural model technology seems to be their inability to monitor what they are doing[3]. Pure statistics and mathematically originated economic modelling also have problems in monitoring and visualising the problem and its solutions. Monitoring and visualisation are essential characteristics in investment decision support systems because they inspire the user and engender a sense of trust.

2.3 State of the Art

Given the requirements discussed in the previous section, it can be said, while looking at the recent research literature ([118], [146], [42], [70], [185]), that hardly any published work fulfills all of them. In this section, we quote some thoughts which illustrate that point. They are meant to provide a cross section of work being undertaken rather than to give a comprehensive review.

Econometrics. GRANGER, (1999), [70], in his book "*Empirical Modelling in Economics*" states that the purpose of the modelling exercise of parts of econometrics is to find a model that is well estimated and appears to fit the data well. It does not, however, guarantee that the model will also be useful for a decision maker. Mainly there are two reasons why that happens. Firstly, econometric models are often over-parameterized and thus they do not perform qualitative - generalized - forecasts that can be used for decision making. Secondly, they are restricted to relatively small data sets and their extrapolation properties are poor. CLEMENTS and HENDRY, (1998), [42] pointed out the inability of macro-econometric models to follow structural changes. Non-constancies in

[3]ZAPRANIS and REFENES [185] have done some remarkable work in order to treat ANNs as statistical devices for non-linear, non-parametric regression analysis, rather than as a form of artificial intelligence.

the data set, they said, are responsible for some of the major episodes of predictive failure because of uncertainty due to parameter estimation and lack of generalization. They suggest, therefore, the idea of modelling structural change using dynamic, switching regression models. Some years earlier, NELSON and FOSTER, (1992), [124], presented a representative econometric model, the univariate ARCH model, and some useful suggestions for its design process. They pointed out two major limitations or, in other words, two widely voiced criticisms against econometric models. Firstly, econometric models are *ad hoc*, i.e. although they can be applied to a particular part of an economic problem, they are not economic models, they are statistics (CAMBELL et al., (1993), [27], ANDERSEN, (1992), [5]). Secondly, in applied work, there is considerable arbitrariness in the choice of econometric models because of the plethora of models. Generally, it can be said that the criticism against econometrics derives mainly from lack of robustness and simplicity in their modelling ability.

Technical Analysis (TA). TA based automated trading systems can be used as just one additional indicator in the overall decision-making process by alerting investors and traders for trend reversals. However, they heavily suffer from lack of adaptivity to dynamic market condition changes. Furthermore, TA trading rules also suffer from subjectivism in their interpretations. It has been said in the literature that "*TA is the art of interpreting a number of reliable and scientifically derived indicators*" (PRING, (1998), [143]). The question that arises out of this definition is whether a system based on its designer's interpretation skills, could actually be robust and be trusted for global investment actions. On the other hand, TA derived automated trading systems are simple systems which can satisfactorily visualise market trends and monitor the trading rules applied. TA theory, therefore, is well understood as well as the proposed trading positions. Additionally, TA rules are easy to implement and they are cheap in terms of computational power and time (MURPHY, (1999), [121]).

Computational Intelligence (CI). One of the first attempts to employ computational intelligence in finance was made when LAPEDES et al., (1987), [101], explicitly used Artificial Neural Networks (ANNs) for modelling deterministic chaos. From those early stages, it became clear that CI and thus ANNs will not be the expected panacea in financial modelling. WHITE, (1988), [180], in his effort to forecast IBM daily stock returns, reported that ANNs, despite their universal approximation properties and their ability to operate with little pre-knowledge in the application domain, suffer from: practical problems in the architecture design process, difficulties in deciding on the correct form of their learning, an inability to explain their underlying mechanism. After more than a decade, REFENES et al., (1999), [147], reported the fact the ANNs had still not convinced practitioners and statisticians of their effectiveness. This is broadly explained by the lack of systematic tests of statistical significance for the various parameters that are estimated for the models. Without any doubt there exist strong links between White's (1988) and Refenes et al.'s (1999) main problems. In the meantime, serious research has been published on computational intelligent financial applications, including not only ANNs. KIMOTO et al., (1990) [95], reported excellent profit margins when they applied ANNs for generating *buy* or *sell* stock signals. However, the evaluation period of their tests was only two years. Contributing to the ANNs validation procedure, is the work of WEIGEND et al., (1990), [177] and MOODY et al., (1992), [118, 116]. Their effort is targeted at achieving optimal network design, training rules and parameters using extra architecture complexity penalising models and more realistic performance functions, such as prediction risk. ABU-MOSTAFA, (1993), [1] in one of his papers in the "Neural Networks in the Capital Markets" conference series, introduced a method of injecting *a-priori* expert knowledge during the ANN's training phase. His approach is basically based on complementary information derived when ANN models are applied on reciprocal problems, like the prediction of the US-Dollar/DeutscheMark and the DeutscheMark/US-Dollar exchange rates. Mostly, this interesting approach indicated the importance of using *a-priori* knowledge to improve the network's generalization properties together with its robust functionality. The issue of integrating expert knowledge still

remains open in financial modelling. PORTUGAL, (1994), [139], investigated through an empirical exercise the performance of ARIMA models against Unobservable Components Models (UCM) and Artificial Neural Networks (ANN) in forecasting monthly Brazilian gross industrial output data. He claimed promising ANN performance particularly in large forecasting horizons. KINGDON, (1997), [97], systematically summarised studies on financial modelling-forecasting-trading and argued in favor of the computational intelligent approaches. According to Kingdon, tests on 501 time series, both of simulated and real nature, showed that ANNs outperform the classical regression econometric methods (see BOX-JENKINS [19]; HILL et al., (1992), [73]; MAKRIDAKIS et al., (1982), [110]; SHARDA et al., (1990), [160, 161]; TANG et al., (1990), [169]; SASTRI et al., (1990), [156]; FOSTER et al., (1991), [58]; WEIGEND et al., (1992), [178]; SHAZLY et al., (1997), [162]). Other techniques used for financial modelling, forecasting and trading, and assigned to the computational intelligence domain are Genetic Algorithms (GA) (see GOLDBERG, (1989,1991), [66, 67]; DAVIS, (1991), [45]; STENDER et al., (1994), [167]; FELDMAN et al., (1990), [57]; KINGDON et al., (1995), [96]; DEKKAR et al., (1994), [47]; GOONATILAKE et al., (1994), [69]), Reccursive Modelling (PESARAN et al., (1994), [131]; (1995), [132]; (1998), [133]), Hidden Markov Models (see PAPAGEORGIOU, (1997), [129]; WEIGEND et al. (1998), [179]). There is an increasing level of interest in applying these techniques in finance for modelling, forecasting and trading but their detailed investigation does not fall within the scope of this work.

Pattern Analysis. Early in the 1990's, R. AGRAWAL [3] started researching on queries about similarity model development in sequence databases. Particularly in 1993, [3], he proposed a similarity search scheme based on R^*-trees (BECKMANN et al., (1990), [12]). According to his work, all the similarity measures are made in the frequency domain by employing *Discrete Fourier Transform* (DFT) (OPPENHEIM et al., (1975), [127]) and making use of the *Parseval*'s theorem (RORABACH, (1997), [149]) which makes *Euclidean Distance* measures also applicable in the frequency domain. In spite of the satisfactory performance, his method is computationally expensive because of the frequency trans-

formations. Additionally, working in the frequency domain does not allow the user to project the results straight back to the time domain so as to easily monitor the model's performance through time, as required when processing financial data. FALOUTSOS et al., (1994), [54], extended the method described above, to fast subsequence matching. The application of their approach on stock price movements has been proven computationally efficient. GOLDIN et al., (1995), [68], introduced a group of transformations which made the distance metric of similarity more invariant to noise and the matching process more robust. However, in financial time series there are non-matching gaps which interfere within similar time series patterns. These are due to abnormal financial events, translation and noise. AGRAWAL et al., (1995), [4], and SRIKANT et al., (1995), [164], dealt with that matching problem and discovered several interesting economic matches. BANAVAS, (1999), [8], discussed a flexible graph matching approach used for forecasting, which also identifies slightly distorted patterns in financial time series.

As can be seen from the presented literature survey, hardly any method exists that combines all requirements for a successful forecasting-trading system. A combination of those requirements is attempted in this research.

Chapter 3

Growing Neural Gas

"It is far better to foresee
even without certainty
than not to foresee at all."

Henri Poincare,
in The Foundations of Science, page 129.

3.1 Introduction

One of the major problems when employing neural network technology in particular economic prediction tasks, is the selection of the network architecture. Apart from selecting among different neural network technologies, there are also a huge number of different architectures that can be used. There are statistical techniques that can be applied to pick an efficient network architecture (see ZAPRANIS et al., (1999), [185]). However these techniques require first the limitation of the searching space, second optimised searching algorithms, and third extensive computational time. Additionally, fixed neural network architectures do not guarantee optimum performance throughout the input data set. In other words, dynamic change in the underlying structure of the data cannot be captured by fixed network structures because of the fact that they are fixed. In this chapter, we

test the forecasting performance and the trading profitability (positive or negative) of a neural network which is characterised as *growing*. This network is called the **Growing Neural Gas** (GNG) (see FRITZKE, (1995), [60]) and it mainly adapts itself not only in terms of calculated forecasting error but also in terms of number of nodes. It automatically adds and removes nodes from its architecture according to the complexity of the problem and the change of the underlying structure of the data. In the following, we describe the algorithm that implements the Growing Neural Gas Network and analyse its performance both as a predictor and an auto trader.

3.2 Growing Neural Gas (GNG)

The neural model we are about to describe was first conceived by MARTINETZ, [113] and extended by FRITZKE, (1995), [60], who added the growing component of the algorithm. The "Neural Gas" algorithm [113] was first applied to the vector quantization, data compression problem and it has been shown, that it 1)converges quickly to low distortion errors, 2)reaches a distortion error E lower than that resulting from K-means clustering, maximum entropy clustering (for practically feasible numbers of iteration steps) and from Kohonen's feature map, and 3)at the same time obeys a gradient descent on an energy surface (like the maximum-entropy clustering in contrast to Kohonen's feature map algorithm).

The incremental component to the above mentioned "competitive Hebbian learning" (CHL) / "Neural Gas" (NG) combination, has brought a number of advantages. Firstly, only a small number of constant parameters are used. Secondly, incremental models are in a better position for handling non-stationary distributions. Generally, non-stationary distributions are a problem in classical neural network approaches. In approaches like the Multi Layer Networks, the Self-Organizing Maps, or the Radial Basis Function (RBF) networks, once the adaptation strength has decayed, the network is "frozen" and thus unable to react to subsequent changes in the signal distribution. This was avoided in the GNG by introducing the removal of "*dead*" neurons in the network. Sometimes many

units of the GNG may get stuck in regions of high probability density introduced to the network sometime ago, but which do not represent the current distribution. These units are referred to as "*dead*". Removing such units increases the network's data tracking ability while decreasing the learning error too. However, the "*dead*" units that represent regions of the data with high probability density, may be used as a kind of system's memory that could be valuable during the learning process everytime that the data distribution returns to previous probability stages. Because non-stationary distributions can be found in many technical and in almost all economic processes, this makes GNG a very valuable tool for modelling and forecasting financial distributions.
In the next section the GNG algorithm is described. However, the original sources of the algorithm can be found in [62].

3.2.1 The "Growing Neural Gas" Algorithm

In this report we consider networks consisting of

- a set A of units (or nodes). Each unit $c \in A$ has an associated *reference vector* $w_c \in \Re^n$. The reference vectors can be regarded as positions in input space of the corresponding units.
- a set N of connections (or edges) among pairs of units. These connections are not weighted. Their sole purpose is the definition of topological structure.

Moreover, there are a number of n-dimensional input signals obeying some unknown probability density function $P(\xi)$. The main idea of the method is to successively add new units to an initially small network by evaluating local statistical measures gathered during previous adaption steps. The network topology is generated incrementally and has a dimensionality which depends on the input data and may vary locally.
The complete GNG algorithm, exactly as published by B. Fritzke [60] is given by the following:

1. *Start with two units α and β at random positions w_α and w_β in $\Re^n$.*

2. *Generate an input signal ξ according to $P(\xi)$.*

   Remark:

   The network initially has two units, α and β, which are connected by a non-weighted edge. The units are represented by Gaussian functions whose sigma (standard deviation) derives from the mean length of the edges that are connected to it. Therefore, each unit represents a specific region in the input space and is activated to a certain degree when new input signals ξ are presented to the network.

3. *Find the nearest unit s_1 and the second-nearest unit s_2.*

4. *Increment the age of all edges emanating from s_1.*

5. *Add the squared distance between the input signal and the nearest unit in input space to a local counter variable:*

$$\Delta error(s_1) = \|w_{s_1} - \xi\|^2.$$

6. *Move s_1 and its direct topological neighbours towards ξ by fractions ϵ_β and ϵ_n, respectively, of the total distance:*

$$\Delta w_{s_1} = \epsilon_\beta(\xi - w_{s_1})$$

$$\Delta w_n = \epsilon_n(\xi - w_n)$$

   for all direct neighbours n of s_1.

7. *If s_1 and s_2 are connected by an edge, set the age of this edge to zero. If such an edge does not exist, create it.*

8. *Remove edges with an age larger than α_{max}. If this results in points having no emanating edges, remove them as well.*

Remark:

When a new input signal is presented to the network the nearest neighbouring unit moves towards the input vector ξ by a fraction ϵ_β. The movement of the nearest unit also drags down the neighbouring units by a smaller fraction ϵ_n. This adaptive movement gives to the network the tendency to follow the input vectors and thus the data distribution. The age parameter attached to each edge is updated in each iteration step in order to control active and non-active unit clusters. *Old* network clusters are removed from the network.

9. *If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows:*

   - *Determine the unit, q, with maximum accumulated output error.*
   - *Insert a new unit r halfway between q and its neighbour f with the largest error variable:*

     $$w_r = 0.5(w_q + w_f).$$

   - *Insert edges connecting the new unit r with units q and f, and remove the original edge between q and f.*
   - *Decrease the error variables of q and f by multiplying them with a constant α. Initialize the error variable of r with the new value of the error variable of q.*

Remark:

The GNG network constists of two layers. The input and the output layers. For a classification problem whose purpose is to classify n-dimensional vectors in k classes, the input dimension of the network is n and the output one in k. Similarly, for prediction tasks, (eg. one step ahead prediction), if the pattern generation method is the univariate sliding window technique, then the input dimension of the network is l, the length of the sliding window and the output dimension is one. At each adaptation step, using this architecture, the squared error occuring at the

output units is accumulated to a local error variable attached to the nearest to the corresponding input vector unit. That helps the identification of the network regions that have difficulties with the learning task. When the error exceeds some predefined threshold a new unit insertion is forced in order to reduce the overall error and to simplify the learning task by increasing the network's complexity. New unit insertions, if necessary, occur at constant rate λ. This allows the adaptation of the weighted connections that exist between the input and the output units. In particular, the adaptation is realized for each pair of input and output vectors using the delta rule.

10. *Decrease all error variables by multiplying them with a constant d.*

11. *If a stopping criterion (e.g., net size or some performance measure) is not yet fulfilled go to step 2.*

3.2.2 GNG for classification

In the case of pattern classification, the classification error of the network's learning process is the criterion used for unit (node) insertion or deletion. In [61], FRITZKE demonstrated the application of the supervised GNG learning method on a two-class classification problem and tested it against a conventional RBF network approach as proposed by MOODY et al., (1989), [117]. In such a classification problem, the local error variable measure of the GNG algorithm (step 5) leads to a network whose units are centers, usually Gaussians, distributed over the input data. The distribution of the centers may differ considerably from the distribution of the data. In the optimum network architecture large data clusters, easily separated, are represented by fewer Gaussian centres with large *sigmas* (standard deviations) while less separated data clusters are occupied by finer Gaussian centres (see figure 3.1).

This kind of data representation is also attempted by Radial Basis Function (RBF) network architectures. However, as reported by FRITZKE in [61], there are substantial differences between the two methods. The unit architecture of the RBF networks is

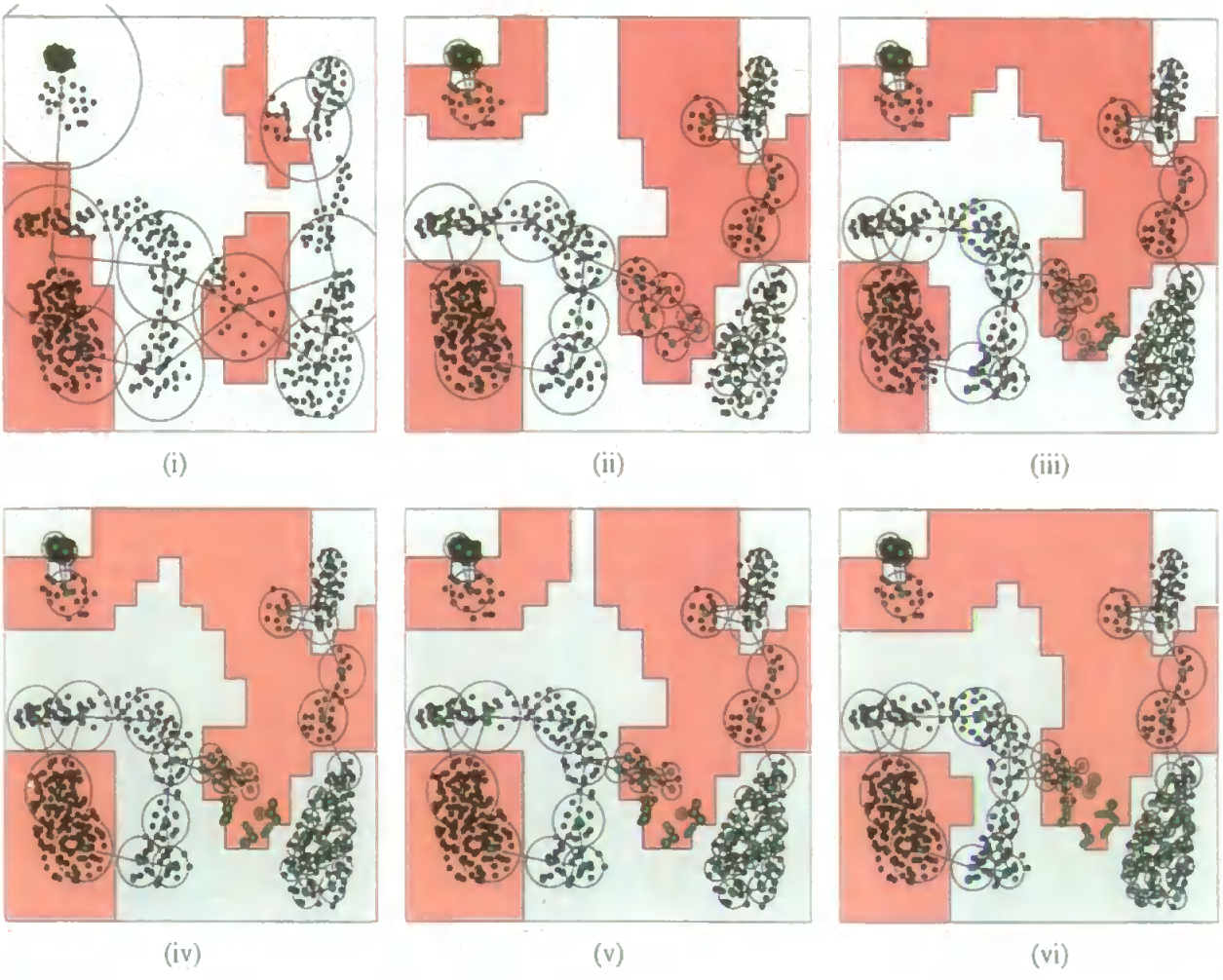


Figure 3.1: Solving a classification problem using the Growing Neural Gas network. **(i)**: 1000 iter. **(ii)**: 5000 iter. **(iii)**: 10000 iter. **(iv)**: 20000 iter. **(v)**: 30000 iter. **(vi)**: 60000 iter. The maximum number of network units that has been used for this simulation was 60. (The pictures have been retrieved using the neural network simulator developed by B. Fritzke)

a-priori fixed and is based on clustering methods (k-means, hebbian clustering) that take into account only the input part of the training data without concerning about the output class labels. In other words, it cannot distinguish areas of the input space which are difficult to classify from ones that can easily be classified. Furthermore, due to the fact that the centres of the RBF networks have fixed positions over the data space, data

points that belong to different classes lying in the region of the same Gaussian activate similar unit vectors. That leads to classification errors as reported by the output units of the network.

On the other hand, the GNG algorithm starts with a small number of units (step 1) and it adapts its parameter set according to the output set of the input vectors (steps 7, 8, 9, 10). Furthermore, the centre positions are allowed to move slightly in order to avoid highly overlapping Gaussian regions and faulty centre positioning (step 6). Finally the accumulated classification error is attributed to the localized unit nearest to the input pattern presentation so that after some adaptive iterations the position of the unit that has more difficulties can be located. A new unit insertion is then forced to cope with this problem (step 5). This can be summarised in the following: "*The network concentrates its resources on the more difficult areas*" (FRITZKE 1996). Unnecessary unit insertion is constrained by the "age" parameter attached to each edge that connects two units. Unused units are deleted after some adaptation steps when they are proved to be inactive. This simplifies the network's architecture and discourages overfitting. Overfitting during the training process is also avoided by introducing stopping criteria of maximum number of units or minimum training error of the classification process. Monitoring of the performance of the network on a separate validation set also helps to avoid overfitting.

With these advantages of the GNG algorithm in mind, its application to prediction, rather than classification, was attempted here. The application domain was financial. The main difference with what has been described above was that the error measures that drive the algorithm for unit insertion or deletion were based on prediction errors. The trained GNG network was then used for predicting financial stock indices. The dimensionality of such prediction problems is usually high and, therefore, the network's topology cannot be easily visualised. The network's capabilities in predicting financial stock indices are presented through statistics on prediction accuracy and trading performance which was measured through a simple trading strategy that incorporates brokerage costs. These results are also compared against those obtained by Multi Layer Perceptron (MLP) network structures.

3.3 Empirical Forecasting and Trading Results using GNG

In this section, we describe the results of applying the Growing Neural Gas (GNG) network to the problem of forecasting financial time series. The network as interfaced to forecasting problems was applied to the following set of European DATASTREAM total market indices[1]:

1. UK-DS-MK
2. FRANCE-DS-MK
3. GERMANY-DS-MK
4. SPAIN-DS-MK
5. ITALY-DS-MK
6. GREECE-DS-MK

The data used in the experiments were daily closing prices for the period 1st January 1990 to the 26th April 2000. From the total of 2693 observations the first 60% (1616 ptns) were used for training, i.e. estimation of the parameters of the GNG network, the following 10% (269 ptns - 12th March 1996 to 21st April 1997) were used for validating the model, and the final 30% (808 ptns - 22nd April 1997 to 26th April 2000) have been employed for out-of-sample testing of the algorithm.

The same experimental framework was applied to the Multi Layer Perceptron (MLP) neural network technology in order to perform one-day ahead predictions. These prediction intervals correspond to daily prediction tasks. The MLP based results were used as a benchmark set in order to evaluate the GNG network performance.

In evaluating the GNG performance, we were interested in answering two questions. Firstly, can the GNG network reach the MLP network's performance and if '*yes*', can it

[1] PRIMARK corporation. [44]

do it in considerably less time? Secondly, did the prediction results of the growing neural network technology we employed correspond to similar results in terms of profitable trading? In this study, we call the answer to the first question *effectiveness of implementation* (e.of Im.) and the one to the second question, *effectiveness of utility* (e.of Ut.). All results are validated against the *random walk* extrapolation method. In table 3.1 the profits obtained through a random walk trading strategy are presented. Note here that transaction costs of 1% have been applied.

	UK	FRANCE	GERMANY	SPAIN	ITALY	GREECE	Aver.
SR	0.67	0.30	-0.45	-0.39	0.34	0.49	**0.16**
Total Profit	3.99	1.76	-1.96	-1.62	4.91	2.96	**1.67**
	Random Walk (1-day ahead)						

Table 3.1: Trading results on European equity indices using the Random Walk extrapolation method. Profit is measured in basis points. SR is the Sharpe Ratio, the annualized return over the testing period divided by the standard deviation of the return series.

Table 3.2 summarises the average out-of-sample performance of the two network models. In this table the directional ability and the trading ability of the models is measured in percentage values and basis points respectively. Every profit (or loss) measure is accompanied by the corresponding Sharpe Ratio risk measure (The mean excess return to the standard deviation of the excess return, [106]). That is the ratio of the annualized profit and the standard deviation of the profit curve. The last row of table 3.2, labeled *time efficiency*, shows the ratio of the execution times that the models need to reach similar performance according to their utility function measures (square mean error). In other words, we measure the relative time needed for the two models to reach similar regions of training error.

	UK	FRANCE	GERMANY	SPAIN	ITALY	GREECE	Aver	
Training Error	0.030	0.027	0.027	0.022	0.042	0.021	**0.028**	G
Directional Ability (%)	51.8	52.2	51.7	50.2	55.1	53.6	**52.43**	N G
Sharpe Ratio	0.86	0.79	-0.46	-0.39	0.83	0.68	**0.38**	net
Acc. Mean Profit	8.49	6.79	-1.63	-2.01	9.89	7.95	**4.91**	class
Training Error	0.029	0.027	0.027	0.022	0.042	0.021	**0.028**	M
Directional Ability (%)	52.7	50.9	51.1	51.6	52.4	51.6	**51.72**	L P
Sharpe Ratio	1.62	0.66	-0.45	-0.37	0.65	0.58	**0.45**	net
Acc. Mean Profit	9.13	6.12	-1.98	-1.57	7.16	6.34	**4.2**	class
Time Efficiency (e.of Im.)	1.16	1.61	1.52	1.97	0.80	2.22	**1.55**	

Table 3.2: Summarised average performance when applying GNG and MLP neural networks for 1-day ahead direction prediction. Trading results (Mean accumulated total profit, and Sharpe ratio) are also displayed. The *Time Efficiency* ratio is calculated as: $TE = (MLPExec.Time)/(GNGExec.Time)$

The results presented in table 3.2 are the unweighted averages of individual models of the same network class. 20 GNG networks were tested in prediction and trading tasks. The first of these models was capable of increasing its complexity up to to 10 units. The

next one up to 15 units and, consequently, the 20th network was allowed to have up to 105 units. The average performance of all the 20 models was then calculated. The same research framework was applied to the MLP network models.

On the European data set used, the directional ability of the networks is on average over the 50% threshold of the random walk model. Furthermore, both the neural network models used are shown to be more profitable than the random walk model. Transaction costs have also been included in the calculations for measuring the profit (or loss) of the neural network traders. However, it is interesting to note that neither GNG or MLP networks managed to convert random walk losses to profits (see profit values for the DAX 30 and the IBEX market indices in tables 3.1, 3.2).

The next step of the investigation undertaken here, was to evaluate GNG networks against MLPs. No significant differences were revealed according their directional ability or their profitable trading capability. As shown in table 3.2 the models perform similarly. However, the GNG based trading system seems to outperform the MLP system when applied on the Greek and the Italian market indices. The Greek stock market faced more rapid growth after 1996 compared to other European countries. In effect, that may be accounted for by the ability of the GNG networks to adapt to new situations by changing their architecture. On the other hand, MLP models seem to perform better in more mature markets like the UK stock market.

What really differentiates the two network types when applied to financial forecasting problems is *time efficiency*. As seen in tables 3.2, 3.3 and 3.4, GNG networks "learn" faster than MLPs. In this experiment, we measured the cpu-time needed by the two different network architectures in order to reach similar training error regions. On average GNG networks converged more than 50% faster than the MLPs.

The next question that concerns the evaluation of the GNG network model is that of the effectiveness of utility. This basically refers to the relation between forecasting and trading results. The aim is to check whether good forecasting results correspond to profitable trading and vice versa. In order to investigate this, we measured the correlation coefficients between the directional ability percentages and the mean accumulated profit

	UK	FRANCE	GERMANY	SPAIN	ITALY	GREECE	Aver.	
Training Error	0.026	0.021	0.020	0.018	0.030	0.024	**0.023**	G
Directional Ability (%)	54.8	53.7	54.0	53.7	56.2	55.4	**54.6**	N G
Sharpe Ratio	1.32	1.61	0.10	-0.56	1.13	1.07	**0.78**	net
Acc. Mean Profit	14.78	11.03	0.42	-1.74	16.37	13.41	**9.05**	class
Training Error	0.026	0.021	0.019	0.018	0.029	0.024	**0.023**	M
Directional Ability (%)	55.7	51.4	54.8	54.4	55.1	53.9	**54.2**	L P
Sharpe Ratio	2.78	1.93	0.36	-0.39	1.15	0.97	**1.13**	net
Acc. Mean Profit	17.31	10.42	1.01	-0.73	13.62	8.33	**8.33**	class
Time Efficiency (e.of Im.)	0.90	1.20	1.27	1.31	0.95	1.73	**1.23**	

Table 3.3: Summarised performance of the **best** GNG and MLP neural networks for 1-day ahead direction prediction. Trading results (Mean accumulated total profit, and Sharpe ratio) are also displayed. On average the **best** GNG network converges to the specific training error faster than its MLP rival.

series. These coefficients for the average results as well as for the best and worst models are presented in table 3.5. Although the prediction and trading series are not highly correlated for any of the models, GNG prediction accuracy seems to be in line with the

	UK	FRANCE	GERMANY	SPAIN	ITALY	GREECE	Aver	
Training Error	0.035	0.030	0.030	0.024	0.036	0.034	**0.032**	G
Directional Ability (%)	49.1	48.8	46.2	46.5	51.1	50.3	**48.6**	N G
Sharpe Ratio	0.51	0.18	-0.50	-0.26	0.44	0.31	**0.11**	net
Acc. Mean Profit	6.77	3.41	-2.07	-2.49	3.78	2.95	**2.06**	class
Training Error	0.035	0.030	0.029	0.024	0.036	0.034	**0.032**	M
Directional Ability (%)	48.7	47.3	48.1	45.3	51.9	51.0	**48.7**	L P
Sharpe Ratio	0.81	0.42	-0.79	-0.28	0.51	0.30	**0.16**	net
Acc. Mean Profit	5.49	4.31	-2.14	-2.11	1.97	3.33	**1.81**	class
Time Efficiency (e.of Im.)	1.48	1.87	2.08	1.79	2.36	1.93	**1.92**	

Table 3.4: Summarised performance when applying the **worst** GNG and MLP neural networks for 1-day ahead direction prediction. Trading results (Mean accumulated total profit, and Sharpe ratio) are also displayed.

increase in trading performance. Such an indication is not visible for the MLP network model. This behaviour might be explained by the effect that only small price changes are correctly predicted.

	Average Model	Best Model	Worst Model
GNG net	0.76	0.74	0.78
MLP net	0.57	0.12	0.41

Table 3.5: Correlation coefficients between the direction prediction ability percentages and the mean accumulated profit series. The prediction and the profit measures have been obtained by applying GNG and MLP neural networks for financial forecasting and trading tasks.

To conclude this empirical evaluation of GNG models, it is necessary to say that although GNG networks have shown positive performance when applied to financial prediction and trading tasks, they have not be proven "impressive" on their average performance. Of the 20 GNG models tested, there exists a "*best*" GNG model, which exhibits better performance both in terms of prediction and profitable trading than the others (see table 3.3). On the other hand, however, the figures are almost reversed when the "*worst*" GNG model is employed (see table 3.4). This effect reveals high variation of the performance in the neural network model set. Usage of individual models for financial forecasting and trading is, therefore, risky and restricts individual model employment. More reliable conclusions in using network models (in this case GNG networks) can be made by considering the average performance of a network models set. That can prevent significant losses. In practice, though, the human factor still remains crucial in selecting the individual models that constitute the model set. Apart from tuning the network parameters, such as the learning rate or the error thresholds, traders must also contribute towards the selection of the appropriate set of models. Such decisions have to be made either *a-priori* or empirically and, therefore, they still remain an open question.

3.4 Conclusion

In this chapter, we have described the Growing Neural Gas (GNG) network (FRITZKE, 1995, [60]). Following the network's successful deployment in solving topological classification problems, we have interfaced the GNG network to solve forecasting problems. In particular, the GNG network has been applied to financial forecasting and trading for the first time.

The basic GNG property of adapting its architecture by adding or removing nodes out of its structure and tracking the complexity of the problem, has been tested in multidimensional financial prediction tasks. Furthermore, in an attempt to evaluate the network's forecasting ability, a simple trading strategy based on the network's forecasts has been deployed. This has been used to measure the actual trading profitability (positive or negative) of the network. In addition, the properties of the algorithm have been tested on real data sets against the random walk extrapolation method and against the Multi Layer Perceptron (MLP) networks. Two additional areas relating to the effectiveness of the network have also been explored. Firstly, the effectiveness of implementation has been discussed. This concerns the network's time efficiency in terms of learning. For the experimental set that has been used in this work, GNG networks learn faster than their MLP rivals. The second area to be tested here was the effectiveness of utility. This has been explored by identifying any relationships between the forecasting and trading performance of the GNG networks when tested on real financial data sets. In this case, GNG forecasts were found to be more highly correlated with the trading profits (positive or negative) than were the MLP ones.

Although the GNG networks have shown positive performance and outperformed the random walk method, their performance cannot be considered as "impressive". As opposed to MLP networks, they have shown similar predictability and profitability on the European stock market indices. Considerable improvement though, has been measured in terms of learning speed.

To conclude this chapter, it must be said that despite testing a set of different parame-

terised GNG networks, no general rules have been found for choosing the best network. The variability in the performance of different networks still remains a problem. Therefore, it is difficult not only to select the best network for a particular task but also to construct a set of networks that, on average, perform well on this task. However, average network models performance seems to be more reliable than individual network model employment. In finance, this distinction may prevent traders from making extensive loses.

Chapter 4

Pattern Matching in Financial Applications

Prediction is very difficult,
especially if it is about the future!

Nils Bohr,
Nobel Laureate in Physics.

4.1 Introduction

In this chapter a simple pattern matching scheme is deployed for both actual price and direction movement prediction of financial time series. The case examined here involves univariate financial time series and direction movement prediction only. The development of the pattern matching forecasting and trading system is based on the scheme that is depicted in figure 4.1. As seen there, there are five stages that describe the overall procedure. Apart from the two I/O stages, there are three main processing stages. The first is titled "*query pattern selection*" and involves the selection of the time series pattern (query pattern) that describes the present situation of the market series. The second processing stage is "*matching*" and refers to the matching of the query pattern onto the market's

time series past activity. The main effort of the "*matching*" stage is to seek historical patterns similar to the query pattern. The identification of similar patterns introduces the third processing stage, '*forecasting*'. The matched historical composites of the query pattern are requisitioned for forecasting. Seen as stimuli that activated some certain time series historical behaviour, they are used to identify this behaviour. We believe that this historical behaviour has forecasting properties. Processing the market's time series activity that follows the historical matched pattern, might recover forecasting information about the current situation of the market. This idea derives from the hypothesis that repeatable patterns exist in the market series, due to common trading actions that have been taken in the past. In other words, if traders make common interpretations of the current situation of the market and those are reflected in their trading actions, then consequently that is recorded by the market as repeatable time series subsequent patterns. Seen from this perspective, empirical matching is essentially prediction.

Consider, for example, a chief economist in a broker company, whose job is to judge the direction of a stock of a specific company. From where are her/his investment decisions, based on the future movement of a specific stock, derived? An answer to this question can be found in E. PETERS' book, [134], "*Patterns in the Dark*". In his book, Peters mentions that, among other analyses, the economist compares what s/he knows of the current situation of the market with past situations, in order to make a prediction. S/He does so, because the future movement of the stock is basically derived from other people's decisions. Therefore, in his effort to anticipate a decision-making process that will be undertaken by another group of people, s/he searches for similarities between current and past financial events. That kind of processing derives from behavioural psychology which states that when faced with incomplete information, we often base our decisions on similarity with past experience ([134]).

The purpose of this chapter is to computationally automate that process. Whilst this is a different approach, it is in line with increased efficiency of the market. By always taking the current situation of the market as input for the pattern matching for forecasting and trading system (PROGNOSIS), we always adapt to new market conditions. Let me,

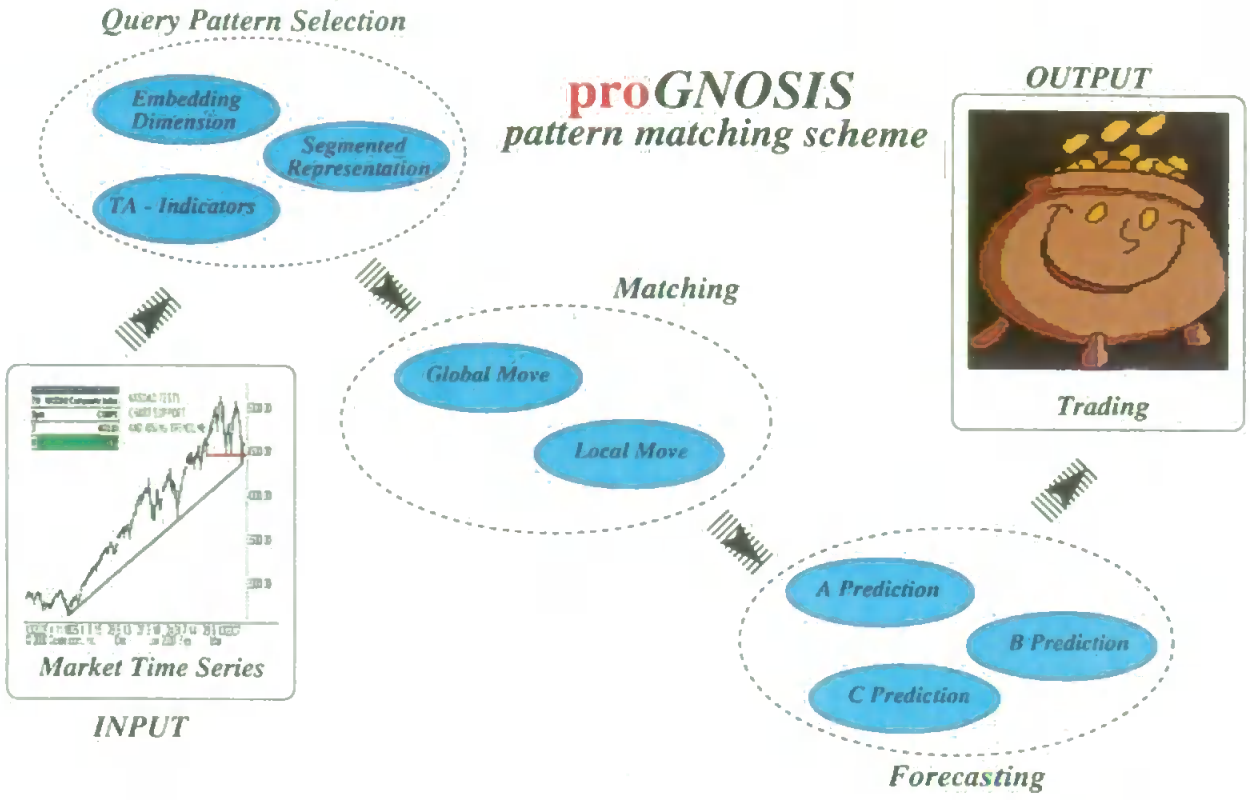


Figure 4.1: The PROGNOSIS Pattern Matching scheme for Forecasting and Trading. Apart from the I/O stages (Input - market series and Output - trading) there are three main processing stages: 1. Query Pattern Selection. 2. Matching. 3. Forecasting.

however, proceed with the technical details of this application of PROGNOSIS.

4.2 Select the Query Pattern

For the model to operate, the query pattern that represents the current situation of the market series, has to be selected. According to the pattern matching system scheme (figure 4.1) this pattern is the one that will be matched onto the market's series past. Here, the subsequent series shape that derives from the last l points of the market series, describes the query pattern. In technical analysis there are certain subsequent series structures that derive from the market's charts. "*Head and Shoulders*", "*Triangles*" and "*Pennants*" are names for technical analysis series formations. A representative collection

of such chart patterns that are classified in the technical analysis literature are given in figure 4.2.

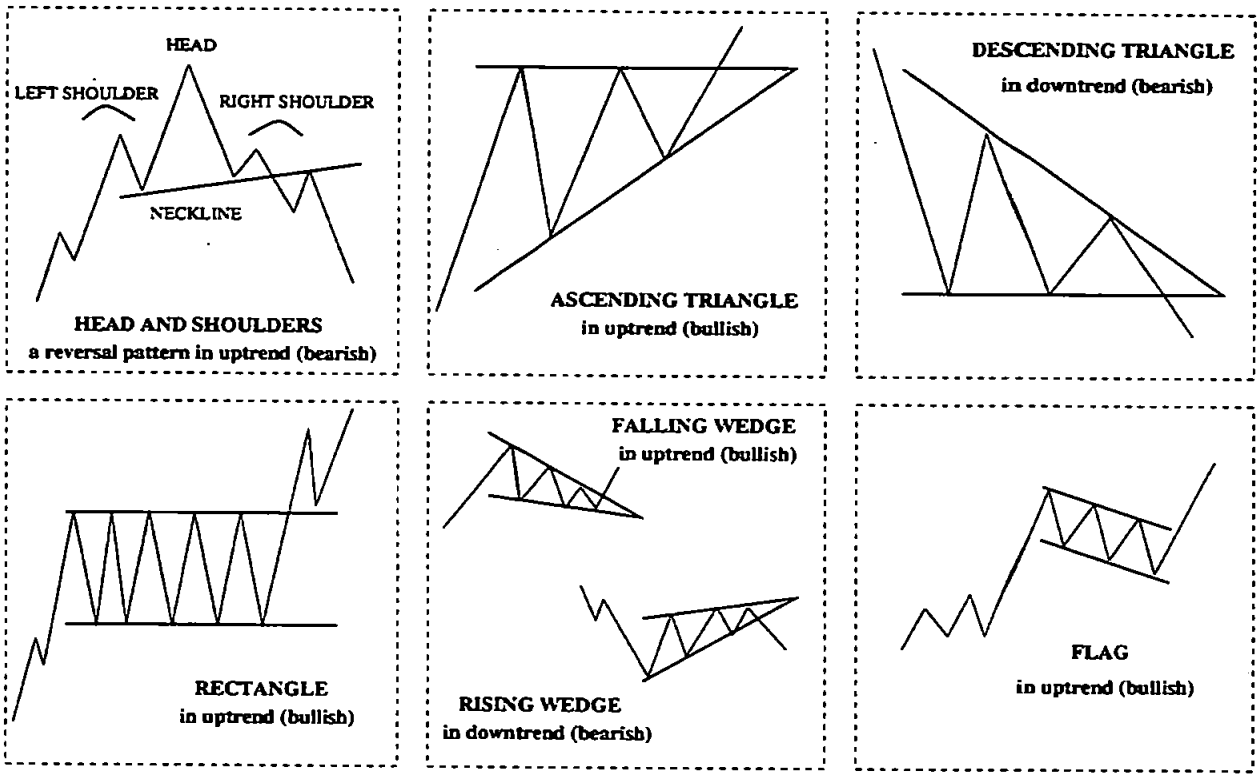


Figure 4.2: Some examples of known chart patterns. **Left Up:** Head & Shoulders. **Center Up:** Ascending Triangle. **Right Up:** Descending Triangle. **Left Down:** Rectangles. **Center Down:** Wedges. **Right Down:** Flags & Pennants. Bullish is a pattern when its last value is greater than its first one. In the opposite situation the pattern is characterised as bearish.

The identification of these pattern formations in the market's charts, if possible, leads technical analysts to certain interpretations for the market's series future development. However, in this work, it is preferred to allow the market series itself to design its own patterns. This is attempted for three reasons. Firstly, the technical analysis chart patterns are limited and are open to different interpretations. Secondly, there may be market series formations that are missing from the technical analysis pattern collection. And thirdly, by letting the market series form its own patterns, new shapes can be retrieved. Additionally, by selecting the query pattern to describe the current situation of the mar-

ket, PROGNOSIS continuously follows the current market conditions and adapts to new ones.

What remains critical is the length, l, of the query pattern. Three methods are proposed here to cope with this problem. A statistical method, which derives from the embedding theory of TAKENS, [168] and SAUER et al., [157], searches for the underlying time series embedding dimension which is used as a measurement of the query pattern's length. Secondly, a method based on technical analysis price oscillator indicators (see [121, 142, 2]) broadly classifies the financial asset time series in bought and sold areas and highlights the most recent crossing point on the time series, where a buy/sell area switch occurred[1]. Thirdly, a piecewise linear segment representation is used for assessing variable lengths of time series sections which are represented by points which on average have the same trend.

4.2.1 Choosing the Optimal Embedding Dimension

To determine the Embedding Dimension (ED) of a time series is a task that has engaged many researchers, TAKEN, (1981), [168], SAUER et al. (1991), [157], and OTT et al. (1994), [128] have developed the mathematical theory for studying a time series system's underlying dynamics. GRASSBERGER and PROCACCIA, (1983), [71], and WOLF et al., (1985), [183], developed algorithms for calculating the correlation dimension and the Lyapunov exponents, respectively, in order to characterise the dynamical behaviour of a time series. In a financial context, LARSEN et al., (1992), [102], computed the correlation dimension for a set of daily dollar rates and reported non-linear deterministic behaviour and BLASCO et al., (1996), [18], identified memory patterns in the Spanish stock market, by employing an embedding dimension searching algorithm.

Our aim here is to find the optimal embedding dimension of the underlying times series model and use it as an indication of the length of the query pattern. A practical method to determine the minimum ED, which does not contain any subjective parameters, does

[1]Price Oscillators are explained in Appendix A

not strongly depend on the available data points, can distinguish between deterministic and stochastic signals and is computational efficient, has been proposed by CAO in [29], [30] and [31]. This method is a close variant of the false neighbour method, described by KENNEL et al., (1992), [91]. In particular, this algorithm iteratively searches for reconstructed pairs of time-delay vectors of dimension d, which remain close according to a distance metric, even when the dimensional space increases by one. Such pairs of vectors are called *true neighbours*. In the opposite case, where the vectors do not remain close in the increased dimensional space, they are called *false neighbours* and perfect embedding exists. The vector dimension d for which false neighbours exist, indicates the embedding dimension which is $d+1$.

The whole procedure is systematically implemented with the following algorithm:

1. Consider a time series $x_1, x_2, \cdots, x_N$ time-delay vector as shown in equation 4.1:

$$V_i(\tau, d) = [x_i, x_{i+\tau}, \cdots, x_{i+(d-1)\tau}], \quad i = 1, 2, \cdots, N-(d-1)\tau \tag{4.1}$$

   τ and d are the time-delay and the system's embedding dimension respectively. τ is set to one ($\tau = 1$), according to the method of mutual information by FRASER and SWINNEY, (1986), [59].

2. Calculate the ratio $D(i, d)$, which is defined as:

$$D(i, d) = \frac{\|V_i(d+1) - V_{nn_i}(d+1)\|}{\|V_i(d) - V_{nn_i}(d)\|}, \quad i = 1, 2, \cdots, N-1, \tag{4.2}$$

   where $\|\cdot\|$ is some measurement of Euclidean distance and nn_i is the index that points to the closest match of the ith time-delay vector.

3. Investigate the variations of the average value of $D(i, d)$ over the time series as calculated by equation 4.3

$$E(d) = \frac{1}{N-1} \sum_{i=1}^{N-1} D(i, d) \tag{4.3}$$

The value of d, for which the fraction:

$$W(d) = \frac{E(d+1)}{E(d)} \tag{4.4}$$

do not change substantially, identifies the optimal embedding dimension of the time series. That is $d+1$.

In spite of the advantages mentioned, Cao's method is sensitive to noise levels, particularly when applied to financial data.

In this work, we employ additional features that incorporate shape and temporal information in the distance metric calculations. The new distance metric is not that sensitive to noise and therefore it can identify more reliable neighbours of the time-delay vectors for the embedding dimension calculation. The shape information is quantified by the slope of the local trend line to which the corresponding vector point belongs. That distinguishes points which have similar values but are part of different trends. Furthermore, the time information is incorporated in the distance metric as a time label which is determined from the position of the vector point in the time series. This distinguishes current vectors from older ones.

The distance metric is given by the following equation:

$$\|V_k(m) - V_l(m)\| = \frac{\alpha \max|x_{k+i} - x_{l+i}| + \beta \max|s_{k+i} - s_{l+i}| + \gamma \max|t_{k+i} - t_{l+i}|}{3} \tag{4.5}$$

where x, s and t are the actual value of the vector point, the slope of the local trend and the time label of the point respectively. α, β, γ are weighting factors that summate to one. The calculation of these additional features does not put much additional strain in terms of computational time and is computationally tractable. The computational complexity increases linearly.

The embedding dimension of the time series, as calculated with the method described above, is used for the query pattern's length selection. It defines the optimal pattern length that has to be matched onto the time series past.

Other methods employed for choosing the optimal embedding dimension are: Singular Value Decomposition (SVD), (BROOMHEAD et al., (1986), [23], PRESS et al., (1992), [141]), invariant on the attractor computation (GRASSBERGER et al., (1983), [71]) and the method of *false neighbours* (KENNEL et al., (1992), [91]).

4.2.2 Technical Analysis - Price Oscillators

Most chartists and speculators agree that *Dow Theory* (NELSON, (1903), [123]), provides a solid foundation for any study of technical analysis. It is, though, very important to note that Charles Dow never viewed his theory as a method for forecasting stock market directions. Its real value, he felt, was as a barometer for reading general market conditions. This is also how we intend to use technical analysis (TA) indicators for identifying how long ago in the recent past the current market conditions were prevalent. Among a set of representative TA indicators, such as the Moving Average Convergence-Divergence (MACD), the Relative Strength Index (RSI), Stochastics Oscillator (StOsc), Moving Average (MA), Price Oscillator (PrOsc) etc. (see ACHELIS, (2000), [2]), we chose to work with price oscillators (PrOsc) in both trending and non-trending markets. PrOscs are derived from the class of momentum indicators and are known for their ability to classify the market in over-bought and over-sold areas. Price Oscillators are usually defined by a three parameters notation, **PrOsc** A-B/C. A and B are the short- and long-term moving averages whose difference produces a series known in technical analysis, as the oscillator. The crossing points of the oscillator series and its C-days moving average are, in the technical analysis literature, the series points where trading actions should be taken. Whenever the moving average curve crosses the oscillator from below a *buy* trading signal is generated. In the opposite case the price oscillator prompts for selling. It is not always the case that the price oscillator trading indications are correct, especially in highly volatile markets. However, they have some power, particularly when their three parameter set is optimised for maximum profit over a testing period. There are studies [53] showing that a price oscillator based trading strategy can consistantly return profits

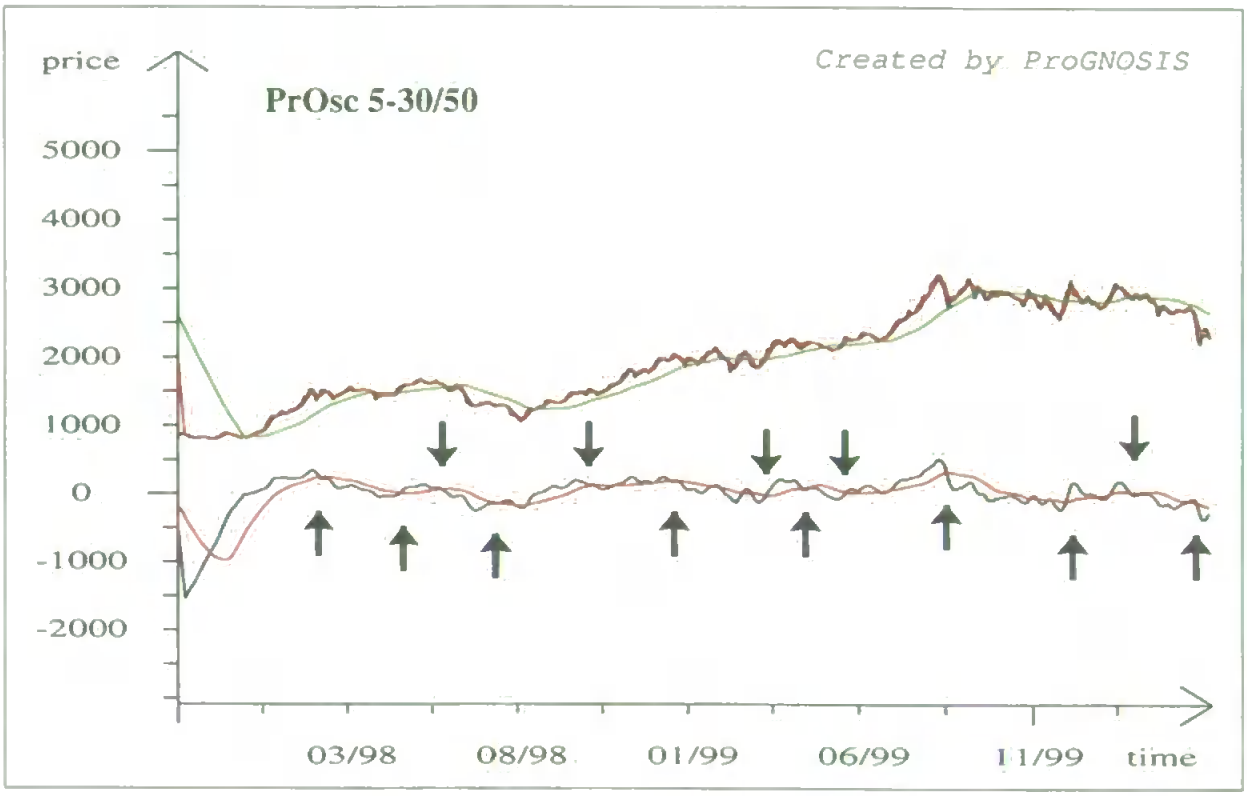
over a significant period of time and for a specific set of parameters A, B, C. For different periods profitable price oscillator trading may be generated for different parameter set. The market series, therefore, is assumed to be separated into *bought* and *sold* regions, when the price oscillator based trading is followed. Figure 4.3 depicts these regions on the Greek market index for the period 01.01.1998 to 26.04.2000 as appeared using the PrOsc 5-30/50 (time series points between two successive arrows).

Here, we use the price oscillator properties to select a suitable query pattern's length. This is performed after the most profitable price oscillator is retrieved for a particular market time series. For this purpose a large number of price oscillators is tested. The moving averages parameter set that returns the maximum profit over the whole testing period[2] is used for the query pattern's length selection. The number of points between the last price oscillator trading signal and the current point of the series identifies the length of the query pattern that will be matched onto the time series.
More details about the development and the advantages of technical analysis price oscillators can be found in Appendix A.

4.2.3 Segment Sampling

Another way of extracting the current pattern is to isolate it from a previous pattern which has a different trend. It is essential for that purpose to represent the data not sampled in equal time intervals, like days, weeks or months, but in variable time intervals characterised by common trend. A nice way of achieving that is to linearly segment the data. There are several ways of segmenting time series data ([56, 39, 90, 165, 166, 40]). Here, we have selected an algorithm proposed by KEOGH, (1997), [92], based on hierarchical segment time series representations (A detailed study about this representation is given in the next chapter). Having segmented the data, we end up with representations like those shown in figure 4.4. In segmented financial time series the length of the last segment indicates the time that the "current" financial situation is following a common

[2]In the profit calculation brokerage costs of 1% over the transaction price are also included



(↑ ↓) point to positions where trading signals are generated.

— Upper black line: Actual time series
— Upper red line: Short-term MA
— Upper green line: Long-term MA
— Lower black line: Price Oscillator
— Lower red line: Price Oscillator's MA

Figure 4.3: An example of a price oscillator (PrOsc 5-30/50) applied on the Greek market index (01/01/1998 - 24/04/2000). The arrows on the graph point to trading signals generated from the price oscillator indicator. Buy and sell signals alternate through out the market series and they segment it in bought and sold areas. Buy signals are generated when the PrOsc curve crosses its moving average from below. The opposite situation indicates a sell signal.

trend. This time interval indicates the length of the query pattern, which is going to be matched onto the time series' history. The advantage of this approach is that it is based on data driven segmentation rather than on any subjective linear trend line tracing.

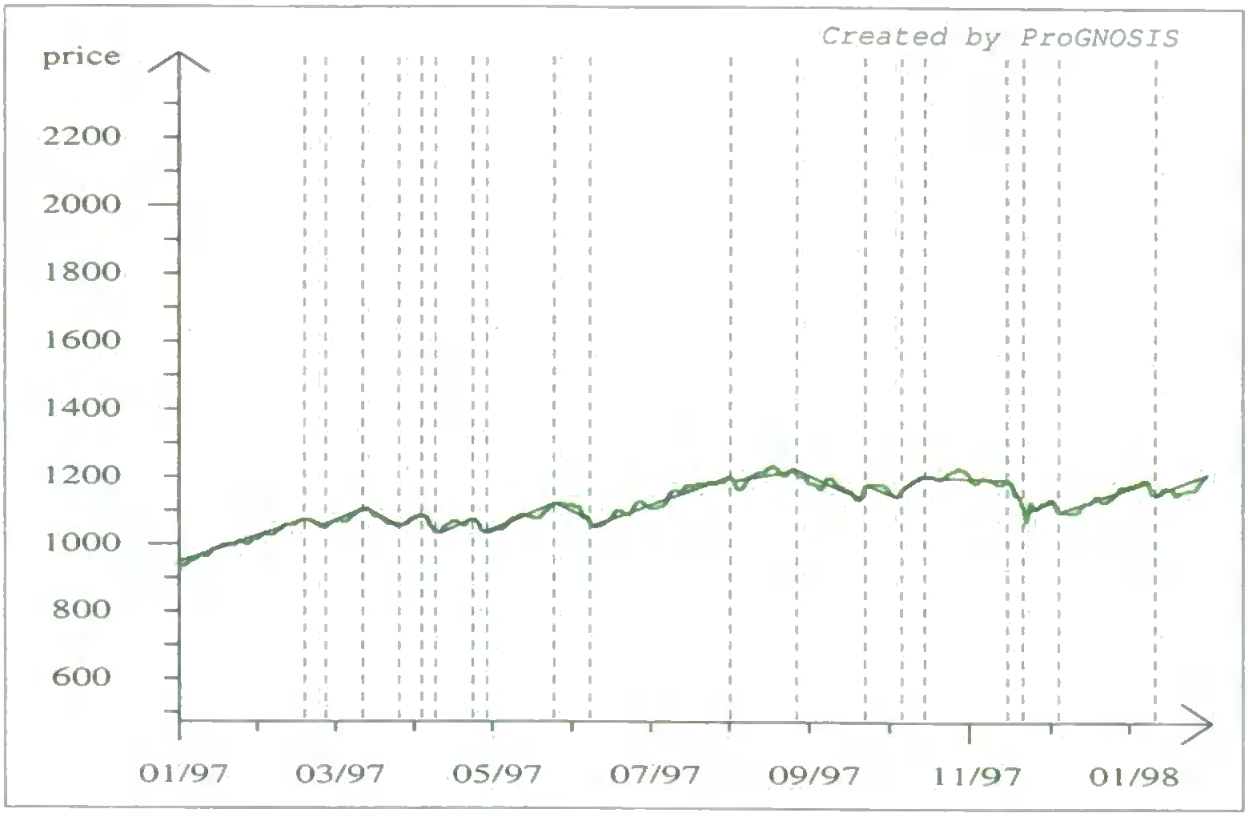


Figure 4.4: A segmented representetation of the CAC 40 French market index (January 1997 - February 1998). The automated segmentation algorithm introduces a new time sampling of the time series which is based on the time series local trends-segments (see black and blue dashed lines). The length in data points of the last segment suggests a possible length for query pattern to be matched onto the financial time series.

4.3 Graph Matching

The pattern matching stage of the system depicted in figure 4.1, involves either the matching of patterns picked up from a database of already classified, technical analysis patterns (see selective representative patterns depicted in figure 4.2) or the matching of patterns that describe the "current" financial time series situation, as selected with one of the methods described in section 4.2. In this thesis, we will work with the latter option. Therefore, the purpose of the matching algorithm, we are about to describe, is to identify, in the time series, historical occurrences similar to the query pattern.

The matching algorithm consists of two phases: **Graph Extraction and Matching**.

4.3.1 Graph Extraction

Graphs are representations of subsequent time series (TS) parts. In PROGNOSIS, "current" patterns are portrayed with two dimensional *Elastic Graph* (EG) topologies, which capture the pattern's shape, value and time duration attributes. *Elastic Graph Matching* (EGM) techniques basically originated from a neurally inspired object recognition architecture (LADES et al., (1993) [100], WISKOTT et al., (1995-1996), [181, 182], TRIESCH et al. (1997), [172]), which is based on local image descriptions - GABOR wavelet transforms - and geometric constraints. EGM has been very successfully applied to face, gesture and object recognition tasks. We will extend the EGM idea to time series recognition tasks and introduce a set of appropriate feature descriptions. We will also show EGM's performance on the recognition of financial time series. The correctly recognized patterns (patterns with high similarity values) will be used for prediction.

An elastic graph is composed of a set of nodes, weighted with time series data point features, which are successively connected with edges weighted with distance measures as imposed between two successive nodes. In figure 4.5, graphs are positioned on subsequent time series patterns. As depicted there, the extracted graph model constitutes a compressed representation of the time series pattern, from which noise distortions and high frequency fluctuations are excluded. Its elastic attribute allows the matching of patterns distorted on the time axis because of local time shrinks and stretches. Each graph node is weighted with features which locally describe the time series point it is placed on. The feature set is composed of the magnitude value of the corresponding time series point, the slope of the edge that connects the node with the following one and the time duration until the next node appears. Correspondingly, each edge is weighted with its geometrical length. The number of nodes used for a graph representation varies between 15% and 25% of the actual data points that represents. Extensive evaluation of different graph representations showed that graphs with this range of number of nodes can describe the

shape of the query pattern. The reduction of the graph nodes compared to the actual number of data points that represents results to a speed-up in the similarity computation that ranges between 75% and 85%.

A drawback of the graph model described above, is that it has to be extracted manually. For now, the injection of a priori knowledge, seen as expert graph extraction, has been judged as necessary. At that stage, we preferred to lay down the benefit of automization for graph model reliability. The automatic extraction of graph models is of great interest and could be pursued in further work.

4.3.2 Matching

The next step is to match the extracted graph model onto the time series. The purpose of matching algorithm is basically to position the graph model onto a set of node positions x_m of the time series. The search strategy for the nodes positions must simultaneously obey two constraints: The subsequent time series pattern information attached to each node must match the time series position information, where the node is projected. The edge lengths between the matched node positions must not differ substantially from the original edge graph lengths. These demands are mathematically expressed by a similarity function for the nodes and a cost function for the edges. To each node, a set of three time series features is attached. The feature set is defined in equation 4.6.

$$I^n = \{Price^n, Slope^n, Time^n\}, \qquad n - node\ index. \tag{4.6}$$

The similarity function compares the node information attached to each node with the corresponding information at each time series point where the node is positioned. The similarity is calculated by taking the normalized average of the Euclidean distances of the individual features. This is shown in equation 4.7.

$$S^I(I^n, I(x)) = \frac{pS_p + sS_s + tS_t}{3} \tag{4.7}$$

where $p + s + t = 1$. The total similarity, when the N nodes graph G is matched at a nodes position x_m onto the time series, is given by the standard deviation of the similarity

of all the nodes (equation 4.8).

$$S^G = std(S^I(I^n, I(x_m))) \tag{4.8}$$

A total similarity curve is plotted, while scanning the processing time series with the graph extracted from the query time series pattern P^Q. When the similarity curve reaches a local maximum, the graph may be distorted in order to compensate for minor variations in time series pattern shape. A penalty factor to limit the local graph distortions is defined by a cost function C. This function does not allow big elongations or diminutions of the edges of the graph and controls the graph topology, as defined in equation 4.9:

$$C(e) = (\frac{original\ length - distorted\ length}{original\ length})^2, \qquad n - edge\ index. \tag{4.9}$$

The square of the relative change in edges length prohibits large distortions and tolerates reasonably small changes. Again, for the whole graph, the total topological cost is given by the average of the costs calculated for each edge (equation 4.10):

$$C^T = \frac{1}{E}\sum_i C(e_i) \tag{4.10}$$

where E is the total number of edges.

However, the major role in the matching process is played by the qualitative similarity measure as defined in equation 4.8. Extensive experimentation has shown that the role of the topological cost function is minor.

The total similarity of the whole graph structure, matched at a position x_m onto the time series is given by:

$$S^T = S^G - \lambda C^T \tag{4.11}$$

The coefficient λ controls the distortion sensitivity of the graph. Large λ penalizes distortions more heavily. Values of λ varying between $[0.2, 0.3]$ give reliable matching results, as derived from an extensive stock indices evaluation set.

As outlined above the ultimate goal of the graph matching algorithm is to place the graph onto a time series position x, which yields high total similarity. The graph matching algorithm is summarised as follows:

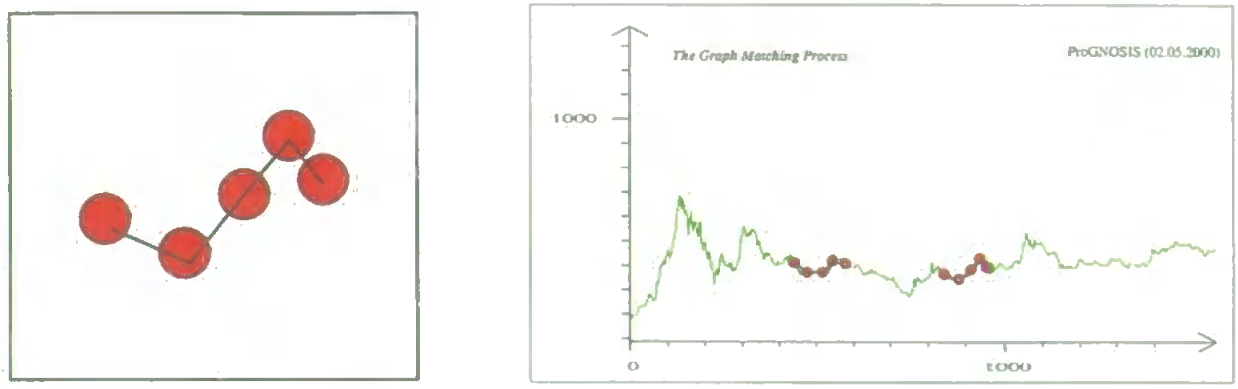


Figure 4.5: Graph Matching: The graph depicted on the left has been extracted from the time series as the graph which describes the current position. The graph has been matched to a similar position on the time series after it has been reasonably distorted (on the right).

1. **Graph Extraction**: Select the time series pattern to be matched. There are three options to select the query pattern. Choosing the optimal embedding dimension, applying the most profitable technical analysis price oscillator, and segmenting the data. A graph is manually extracted out of the query pattern. The graph carries necessary time series features which incorporate both magnitude and shape information.

2. **Sequential Scanning**: The time series is sequentially scanned in coarse steps defined by the time series sampling rate. Daily stock index closing prices, for example, are scanned in daily steps. The similarity measure given in equation 4.8, is used for placing the graph on a candidate position of high similarity. The graph's main structure remains unchanged.

3. **Local Corrections**: Each graph node is allowed to move forwards or backwards, up to three positions. The algorithm, using the topological cost function (equation 4.9), seeks for the optimum node position. After the first node is positioned, the algorithm is applied to the next one. The total similarity is measured by equation 4.11.

4.4 Forecasting

Until now, we have only been concerned with the problem of searching for subsequent time series parts which appear geometrically similar to a recent time series pattern. As outlined in the introduction, in a financial time series, historical parts, which are characterised by movements similar to the current ones, can be used as indications for the time series future activity (Empirical matching is essential prediction).

In this section, we propose a forecasting model which operates with historical matched patterns as derived from the matching algorithm applied on financial time series. This model is adaptive to new market phenomena, comes to terms with the efficient market hypothesis by always considering the 'current' market situation for its calculations and is computationally simple.

4.4.1 Forecasting Model Description

Consider that a set P^H of subsequent historical patterns $p^h(t)$ has been found to fulfill our similarity constraints with respect to the "current" query pattern p^c.

$$P^H = [p_1^h(t_1), p_2^h(t_2), \cdots, p_i^h(t_i), \cdots, p_n^h(t_n)], \quad n = 1, 2, \cdots, N, \tag{4.12}$$

where each pattern p_i^h is of duration t_i, $t_i = 1, \cdots, T_i$. All historical patterns in equation 4.12 are sorted with respect to their similarity value, i.e. $S^{p_1^h} \geq S^{p_2^h} \geq \cdots \geq S^{p_n^h}$. Where more than one historical pattern exhibits the same similarity, they are then sorted from the most recent to the oldest one.

The pattern set P^H is used for the prediction model. α-*days* ahead prediction is performed using one of the following calculations:

1. **Best Matched Pattern**

   (a) Relative to the last pattern value:

$$pred_A = \frac{p_1^h(T_1 + \alpha)}{p_1^h(T_1)} \times p^c(T_c) \quad . \tag{4.13}$$

(b) Relative to the mean pattern value:

$$pred_B = \frac{p_1^h(T_1+\alpha)}{\frac{\sum_{T_1} p_1^h(t_1)}{t_1}} \times \frac{\sum_{T_c} p^c(t_c)}{t_c} \quad . \tag{4.14}$$

(c) Normalized value:

$$pred_C = p_1^h(T_1+\alpha) + \left(\frac{\sum_{t_1} p_1^h(t_1)}{T_1} - \frac{\sum_{t_c} p^c(t_c)}{T_c}\right) \quad . \tag{4.15}$$

2. **Whole Matched Pattern Set**

(a) Relative to the last pattern value - averaged:

$$pred_D = \frac{p^c(T_c) \times \sum_{n=1}^{N} \left(\frac{p_n^h(T_n+\alpha)}{p_n^h(T_n)}\right)}{N} \quad . \tag{4.16}$$

(b) Relative to the mean pattern value - averaged:

$$pred_E = \frac{\frac{\sum_{t_c} p^c(t_c)}{T_c} \times \sum_{n=1}^{N} \left(\frac{p_n^h(T_n+\alpha)}{\frac{\sum_{t_n}(p_n^h(t_n))}{T_n}}\right)}{N} \quad . \tag{4.17}$$

(c) Normalized value - averaged:

$$pred_F = \frac{\sum_{n=1}^{N}[p_n^h(T_n+\alpha) + \left(\frac{\sum_{t_n} p_n^h(t_n)}{T_n} - \frac{\sum_{t_c} p^c(t_c)}{T_c}\right)]}{N} \quad . \tag{4.18}$$

To select the appropriate forecasting calculation, the PROGNOSIS pattern matching scheme is applied $(\alpha+1)$-days ago to forecast α-days ahead. All six (4.13, 4.14, 4.15, 4.16, 4.17, 4.18) forecasting mathematical sentences are used. The most successful one, i.e. the one that gives the minimum squared forecasting error is selected as the "winner" and is the one used for the actual forecasting model.

The employment of this set of linear prediction calculations seems compatible with the notion of different market trends in terms of future movement. We believe that a financial asset time series can be classified in periods that are sometimes characterised by great efficiency, others where it follows the most recent trend and finally others where history simply repeats itself. Our effort is to capture this effect using the forecasting set composed by the equations 4.13, 4.14, 4.15, 4.16, 4.17, 4.18.

4.4.2 GNG - Forecasting

In chapter 3 we described how Growing Neural Gas (GNG) networks can be used for forecasting univariate financial time series. There, several experiments were conducted using the sliding window method in order to extract the input-output patterns from the time series, necessary for the supervised neural network training. Here a more selective supervised training procedure is followed. The training set that will be learned by the GNG network is decided upon by the matching algorithm that has been described in the previous section. What the I/O patterns of this training set share in common is that they are similar according to the similarity function defined in equation 4.7. All the historical occurencies that are similar to the query pattern are used together with their output values to train a small GNG network so that it can perform forecasts. The reasons for choosing the GNG network architecture for the forecasting task are mainly the following:

- Firstly, due to the fact that the number of the historical time series patterns that exceed the 70% similarity threshold does not remain constant, networks of varying complexity are needed. GNG networks, as mentioned in chapter 3, can control their complexity according to the complexity of the problem that they are due to solve.

- Secondly, there is some evidence the GNG networks converge to low training errors faster than their Multi Layer Perceptron (MLP) rivals (see section 3.3). Due to the fact that for different query patterns of variable length, different neural network architectures need to be trained, speed in the network's training process is vital.

For the reasons above GNG networks are being selected at this point to accomplish the forecasting task. The one-day ahead GNG forecasting procedure is summarised in figure 4.6. The forecasting scheme that is depicted there can easily be generalized to α-days ahead forecasts ($\alpha > 1$).

Experiments on the GNG - forecasting approach are conducted in section 4.6 and they are presented in parallel to those based on the forecasting method that has been previously described (see section 4.4.1).

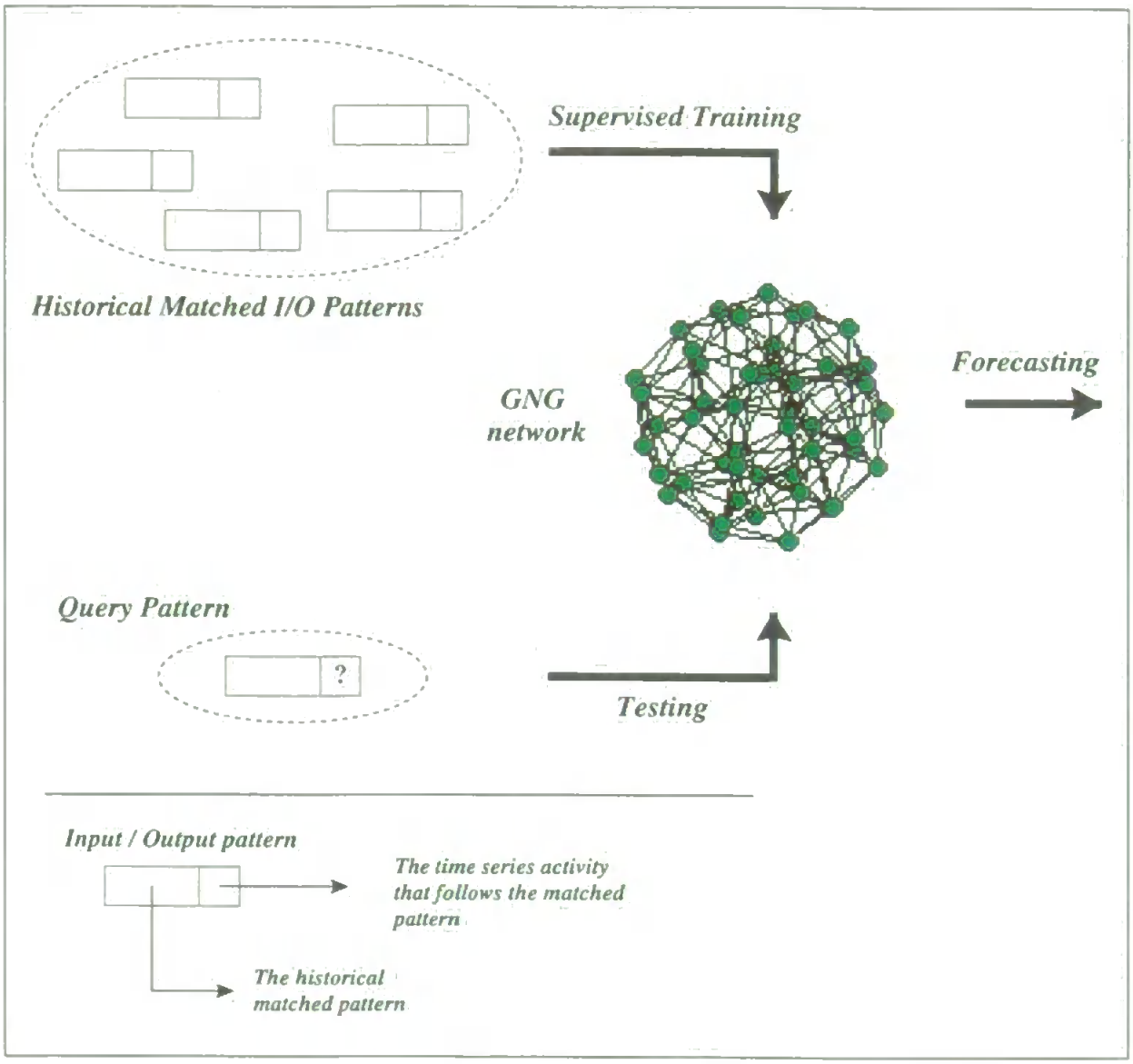


Figure 4.6: The GNG-forecasting scheme. All matched patterns together with their next day values are used to train the GNG network. The query pattern then is the input of the GNG which will produce the next day forecast.

4.5 Trading

Trading system development is a domain in which much work has concentrated. Different methodologies and investment strategies have been proposed, [173, 22, 38, 126, 65], to help brokers achieve maximum profit through their financial activities. An extensive

review on trading strategies is given by CONRAD and KAUL, (1998), [43]. The authors of this review article claimed that less than 50% of the 120 trading modules they implemented yielded statistically significant profits. But what is a trading system in terms of computation? Basically, a trading system is a hybrid of two main modules. A prediction module, which optimises its performance to output accurate predictions and a trading module that, according to a trading strategy and the predictions originating from the prediction module, goes *long*, *short* or *holds* in the market. Other types of trading systems have been proposed in [13, 88, 119]. BERGIO, (1996), [13], KANG et al., (1996), [88], and MOODY et al., (1996), [119] proposed merged variations of the prediction and trading modules in order to overcome the problem of taking trading positions using not profit but prediction criteria. However, they still trained their learning machines with data used for prediction. XU et al., (1997), [184], on the other hand, built a system which learns past investment decision signals and outputs desired investment positions. More recently, MOODY et al. (1998), [120], used recurrent reinforcement learning (RRL) algorithms to train labelled trading data. For his experiments, he used profit and Sharpe ratio performance functions during the RRL. Both applications, according to his article, outperformed forecasting based trading modules. In the same year, TOWERS and BURGESS, (1998) [171], employed parameterized decision rules to choose trading actions derived from a mispricing[3] forecasting model. Intelligent trading systems, which implement sophisticated trading rules may be of great interest but we prefer in this research to follow a simple trend following strategy. This makes our system's performance independent of sophisticated trading rules and simultaneously comparable to trading performance claims derived from fundamental, technical and computational analysis.

One major strategy employed by many stock and futures traders is the use of trend as an aid in making trading decisions. The origin of this behaviour is located in Dow theory, [123], where it is assumed that the market moves in trends which give profitable trading opportunities. Traders usually want to take positions in early trend stages and maintain their positions until trend reversal will occur or will be predicted. In PROGNOSIS such

[3]Significant work on mispricing forecasting models has been undertaken by BURGESS in [24, 25, 26].

a simulated daily trading strategy is attempted. We buy (sell) on the previous closing price if a rising (falling) trend is forecasted. If no change in the trend from the previous day is predicted, the current position is maintained and unnecessary brokerage costs are avoided. This trading rule is summarised as follows:

$$if \quad P(t-1) < P(t) \; and \; P(t) > P(t+1) \quad then\ sell \; - \; "Go\ Short" \quad . \tag{4.19}$$

$$if \quad P(t-1) > P(t) \; and \; P(t) < P(t+1) \quad then\ buy \; - \; "Go\ Long" \quad . \tag{4.20}$$

$P(t-1), P(t), P(t+1)$ are the closing prices of yesterday's, today's and tomorrow's (forecast) respectively[4]. Trading costs are also considered to be 1% of the transaction price. LOEB, (1983), [107] and WAGNER et al., (1993), [176] note in their studies the importance of trade-execution costs. Although recently, CHRISTIE et al., (1994), [41], HUANG et al., (1996), [78] and BESSEMBINDER et al. (1997, 1998), [15, 16], suggested that trading costs may depend on the structure of the market, in the present research we subtract constant trading costs of 1% of the transaction price.
Finally, a stop loss criterion is considered for the system. Every time that losses exceed a 5% threshold, open trading positions are closed in order to prevent extensive losses and the automatic trader waits for a new trading signal to take action. The profit achieved is accumulated over the testing interval. Both the accumulated profit curve and the final profit achieved are displayed by the system.

4.6 Experiments

We experimented with the system in order to find out how it responds to real forecasting environments. The data evaluation set that has been used to run several experiments is composed of six European market indices that have been obtained from the DATASTREAM, [44], database. Those are the UK-DS-MK, FRANCE-DS-MK, GERMANY-DS-MK, SPAIN-DS-MK, ITALY-DS-MK, GREECE-DS-MK. The time window for these series extends from

[4] $P(t+1)$ is replaced with $P(t+\alpha)$ for α-days ahead predictions.

01.01.1990 to 26.04.2000 sampled daily - 2693 points for each series. For consistency the experimental framework is similar to the one described in chapter 3 and is repeated here for completion. 60% of each data series has been used to build its history and the remaining 40% has been used for testing. The test points, within the testing samples set, have been taken randomly. A normal distribution random number generator has been employed for that purpose. 5000 experiments (testing samples) have been conducted for each index time series. The systematic evaluation framework produces results for all three query selection methods (Minimum Embedding Dimension (MED), section 4.2.1, Optimum Price Oscillator (PrOsc), section 4.2.2, and Linear Segment Representation (LSR), section 4.2.3) and for the two forecasting methods (Forecasting Sentences (FS), section 4.4.1 and GNG Forecasting (GNG-F), section 4.4.2). The forecasting results are given in table 4.1. They involve the one day-ahead direction movement prediction of the UK DATASTREAM market index. The rest of the results that correspond to the remaining European indices are presented in Appendix B (see tables B.1, B.2, B.3, B.4, B.5).

To interpret the results shown in these tables and to draw global conclusions is a difficult task. However, apart from the fact that on average the directional ability of the pattern matching system exceeds the 50% random walk threshold, it can be said that the system that combines the minimum embedding dimension (MED) query pattern selection technique and the forecasting sentences method (section 4.4.1) performs better than the others. The statistical origin of the MED technique compared to the empirical source of the remaining two query pattern selection methods, might offer an explanation for this performance. Additionally, due to the variable number of the matched historical patterns, it may be the case that not enough training patterns are presented to the GNG network and, therefore, it lacks in forecasting performance.

Furthermore, the prediction results seem to be more satisfactory when the system is applied to reasonably short-time period time series rather than to longer ones. It can be seen from table 4.1, that the average directional ability of all the 2-years sets is better than the total directional ability over the 10-year period. We suspect that this is due to interference in the forecasting algorithm by the old matched patterns. We believe that

UK-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
MED (%)	57.0	54.4	57.9	64.0	56.0	**57.9**	**55.4**
PrOsc (%)	56.5	52.6	59.1	48.6	49.1	**53.2**	**52.9**
LSR (%)	56.4	54.9	56.6	52.1	48.9	**53.8**	**53.6**
Forecasting Sentences (FS)							
MED (%)	54.1	50.8	50.6	52.9	53.3	**52.3**	**51.9**
PrOsc (%)	53.0	51.1	50.3	47.9	48.2	**50.1**	**50.4**
LSR (%)	52.3	50.5	45.3	49.7	46.5	**48.9**	**48.1**
GNG Forecasting (GNG-F)							

Table 4.1: Testing the forecasting directional ability of the pattern matching system. The UK-DS-MK index case. MED, PrOsc and LSR stand for Minimum Embedding Dimension, Price Oscillator and Linear Segment Representation respectively. These correspond to the three methods for selecting the query pattern. The standard deviation of these forecasts is close to unity. Order of unit std holds for later forecasts as well.

this happens because similar structures (patterns) that are observed on financial time series have different temporal durations, in particular when old patterns are compared to more recent ones. Such pattern relations cannot be captured by the present pattern matching algorithm and, therefore, old historical matched patterns that have the same temporal duration as the query pattern, produce faulty predictions.

For a more realistic evaluation of the proposed pattern matching system, its profitability is measured through the trading strategy that has been described is section 4.5. Table 4.2 presents the profitability test results together with the corresponding Sharpe Ratio (SR)[5] measures. The trading results for the remaining European market indices are given

[5] $SR = (Annualized\ Return)/(std\ of\ the\ returns)$

in appendix B (see tables B.6, B.7, B.8, B.9, B.10).

UK-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	90-2000
MED	15.96	6.92	6.56	13.74	0.25	**8.69**	**8.3**
	(2.27)	(0.70)	(0.99)	(2.03)	(0.13)	(1.22)	(2.01)
PrOsc	8.76	4.60	6.35	2.25	3.99	**5.19**	**10.40**
	(1.26)	(0.21)	(0.91)	(0.33)	(0.81)	(0.70)	(1.81)
LSR	9.45	2.94	-0.49	0.43	-4.26	**1.61**	**0.24**
	(1.97)	(0.58)	(-0.04)	(0.08)	(-0.29)	(0.46)	(0.08)
Forecasting Sentences (FS)							
MED	6.85	-2.68	-2.16	2.94	5.51	**2.09**	**0.73**
	(0.73)	(-1.08)	(-0.94)	(0.58)	(0.76)	(0.01)	(0.20)
PrOsc	4.70	-1.67	-1.68	-3.86	-4.98	**-1.50**	**-0.93**
	(0.60)	(-0.85)	(-0.15)	(-0.94)	(-1.08)	(-0.48)	(-0.24)
LSR	0.99	-2.95	-4.25	-2.86	-7.17	**-3.25**	**-0.67**
	(0.11)	(-1.05)	(-0.78)	(-0.45)	(-1.48)	(-0.73)	(-0.67)
GNG Forecasting (GNG-F)							

Table 4.2: Testing the trading ability of the pattern matching system. The UK-DS-MK index case. The values in quotes are the corresponding Sharpe Ratio (SR) measures.

The observations made previously in table 4.1 are not readily confirmed in the profitability results of table 4.2. That is because an additional parameter interferes in the profit (positive or negative) calculations. That concerns the question of whether big price changes are correctly predicted or not. In table 4.3, we give the average over the whole data set of the correct direction movement predictions that exceed a predefined threshold of 5% change. Financial and macroeconomic time series change usually very slowly over

time; even in deep recessions they rarely fall by more than 10% per day. Therefore, a 5% change in the price of a financial index can be considered a big change. The figures in this table, are not significantly above 50%. This is likely to be the reason why the trading results do not exactly correspond to the prediction ones.

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
FS	60.5%	53.7%	52.2%	57.7%	42.2%	**53.3%**	**48.0%**
GNG-F	59.6%	33.1%	41.3%	48.1%	45.3%	**45.5%**	**28.8%**

Table 4.3: In this table, the average percentages of the correct direction movement predictions on the UK-DS-MK market index that exceed a threshold of 5% in price change, are displayed.

In conclusion, however, it can be said that the response of the pattern matching system in terms of realistic trading is profitable overall.

4.7 Extensions

We have demonstrated here how pattern matching can be applied on financial time series for forecasting and for trading. Throughout our experimentation some observations about the drawbacks of the system have been made. In particular, it has been detected that the forecasting performance of the pattern matching system drops when long time series are analysed. In addition, for very long time series, the sequential similarity search also causes problems in terms of computation.

Both these effects play an important role in financial processing. Due to the fact that almost all financial forecasting techniques are based, to a great extent, on past experience, longer processing data series are better for the forecasting system's performance.

Any extension, therefore, of this pattern matching system, should include attempts to cope with these problems. That would allow the system to take advantage of the continuously growing financial databases. Here, we present some keypoints that can attenuate those problems, although they are not guaranteed to solve them.

A compressed representation of financial time series that is based on piecewise linear segments, [92], and retains most of the shape information of the series, contributes to fast data processing. In addition, the Dynamic Time Warping algorithm, [99, 14], when applied to segmented time series representations, allows time warped comparisons of the patterns and, therefore, supports the investigation of the hypothesis that current pattern formations are similar to older ones if some variability in their temporal duration is considered (see [9]). We investigate these ideas systematically in chapter 5. The same financial data sets have been employed for that purpose in order to allow direct comparison with the research that has been undertaken here.

Chapter 5

Applications on Segmented Financial Time Series

Who controls the past controls the future.
Who controls the present controls the past.

-George Orwell

Time plays a very important role in equity index time series. One of the main attributes of financial time series, in general, is the rate at which they have been sampled. Looking back at the literature, researchers apply their models mostly to data sampled with constant rates. They usually refer to hourly, daily, weekly, monthly, or yearly data. The choice of the sampling rate is made *a-priori* depending on what the researcher wants to model. Even in technical analysis, all methods are applied on constantly sampled data. Consider, for example the moving average technical analysis indicator calculation which requires a moving window of constant length. This has a major impact on the system's performance, before the financial analysis even starts. In finance, there are strong arguments that suggest inaccuracies may be the result of constant sampling rates (see DEBOECK, (1994), [46], GATELY, (1996), [64], DELURGIO, (1998), [48]). Consider, for example, that in financial modelling we wish to retain important aspects of the data while eliminating uninformative, or noisy, data. Many simple ways of approximating and

preprocessing data exist. However they require manual selection of fixed parameters, e.g. the selection of a window size, in a data smoothing calculation.

In this chapter, we investigate the hypothesis that the rate at which important events occur in financial time series varies, and that trend changes may be occurring more rapidly now than in the past. In such a case, a fixed data sampling rate is out of the question. We investigated this hypothesis using the method described in the previous chapter applied this time only on piecewise linear segment represented time series. Experiments have shown that the limited graph distortion effect was not capable of capturing the momentum of time changes in equity index time series. On the other hand, more extensive elongation of the graphs worsened the pattern matching performance by allowing differently shaped patterns to appear as similar. Therefore, we investigate a data representation based on piecewise linear segments. The representation is derived from an algorithm proposed by KEOGH, (1997), [92]. It supports variable sampling of the time series and achieves high compression rates. A pattern matching algorithm applied on segmented time series is proposed in this chapter. This algorithm addresses the problem of matching a query pattern P^Q, extracted from the recent past of a financial time series, onto historical occurencies of similar shape but different duration. Past pattern matches are used to predict the time series' future trend activity. These predictions are accompanied by slope and time duration attributes (BANAVAS et al., (2000), [9]). As we believe that financial time series generate their own patterns, which do not always follow the shape of the known technical analysis chart patterns, we set the query pattern to illustrate the "current" situation of the market (see [8, 9, 10]). The linguistic expression "current" automatically inserts a parameter into the pattern matching system, that has to be fitted properly in order to perform successful predictions. To solve this problem, we introduce here the notation of **Minimum Embedding Segment Dimension** (MESD) and a technique calculating it. MESD, measured in number of segments, indicates the number of successive segments which compose the query pattern. MESD calculations are also used as evidence for non-linear deterministic dynamic behaviour of the segmented time series. To my knowledge, embedding dimension calculations have never been applied

on segmented time series representations. The case that is examined in this chapter is the univariate case, where the financial time series are univariate ones. The matching of patterns, in this context, requires a method that can define a distance measure between patterns of different length. The *Dynamic Time Warping* algorithm, introduced by KRUSKALL and LIBERMAN, (1983), [99], and employed by many researchers (eg. [14]), especially in the field of speech recognition, offers a suitable solution to that problem.
By segmenting time series, we achieve a variable length sampling of the original time series, where the sampling intervals are determined solely by the data and the resulting representation shows the times at which the trend in the market changes significantly. On such representations, the pattern matching algorithm captures data structures which have similar shape but different time scaling. This supports the hypothesis that there is a change in the rate of the market's evolution because of factors such as the spread of the computer technology in financial institutions and the huge expansion of the telecommunications capabilities which substantially increased the number of market participants. Figure 5.1, attempts to illustrate this effect.
As mentioned before, trend duration indications are attached to the trend forecasting output of the system. In effect, the trading module of the system, that is described in this chapter, becomes more selective and efficient, in terms of entering and abandoning the market. Finally, all algorithms, which are also integrated into the PROGNOSIS software package, are computationally very efficient because of the great data compression rates that can be achieved.

The proposed system involves the following processing stages:

- Time series linear segmentation.
- Computation of the minimum embedding segment dimension for query pattern selection.
- Pattern matching on segmented representations by incorporating similarity measures based on Dynamic Time Warping (DTW) and Multiple Feature Sets (MFS).

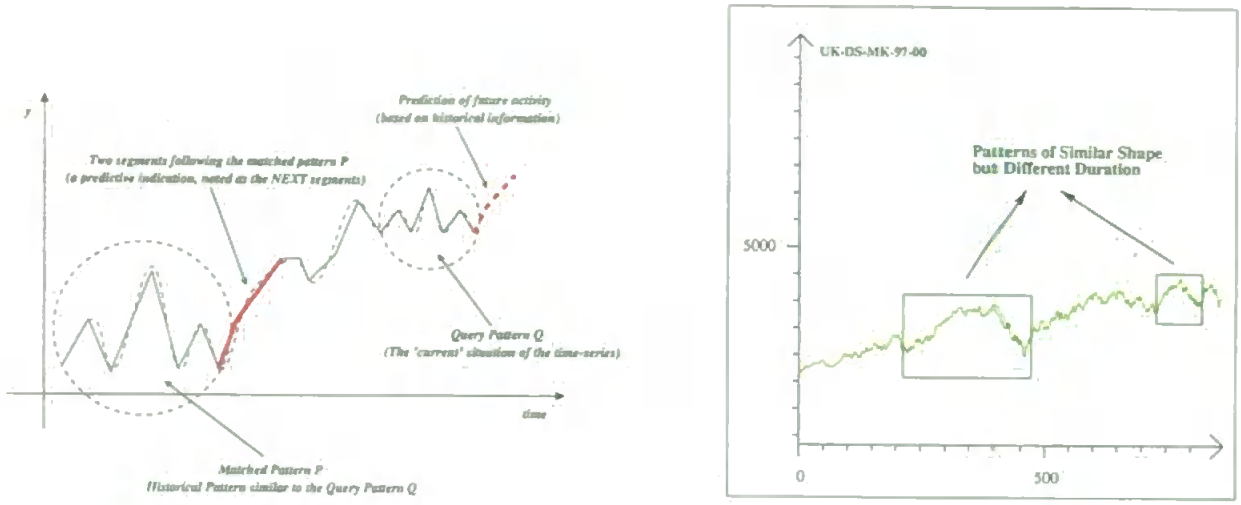


Figure 5.1: The hypothesis: Patterns in finance occurred slower in the past than today, because of computer technology and telecommunication expansion. With conventional matching it is not possible to capture such similarities. **Left:** An artificial example of Head & Shoulders patterns occurring in different resolution. **Right:** We illustrate our hypothesis on the UK DATASTREAM market index from 01.01.97 to 27.04.2000.

- Trend Prediction based on temporal ratios.

- Market index trading based on selective trading rules.

In the following, we briefly describe the time series segmentation problem (section 5.1) and the DTW algorithm (section 5.2). The pattern matching system, as adjusted for segmented time series is discussed in section 5.3 together with its forecasting and trading components. In this section, we also discuss the Minimum Embedding Segment Dimension (MESD) algorithm applied on segmented time series representations. Forecasting and trading results as obtained from tests on market indices are given in the section 5.4, which also includes evidence about the hypothesised change of the rate that "things" happen in financial stock indices. The discussion and outlook section is located at the end of the chapter.

5.1 Segmentation

Time series segmentation can be defined as the process of dividing the data into distinct subsets which have common characteristics. Segmentation is applied in many domains such as image processing, speech recognition and scene analysis. The purpose of segmenting the data in most applications is to allow subsequent information processing using the data subsets. In financial time series, the aim of segmentation is to depict clearly local trends as determined from the data, and distinguish them from noise. The global time series shape must, after a successful segmentation, be retained. Moreover, time series segmentation resembles the way that humans reproduce/draw series with high frequency fluctuations. In figure 5.2 we depict a part of a financial time series, a manual reproduction of it by the author and an automatic segmented representation as produced by PROGNOSIS.

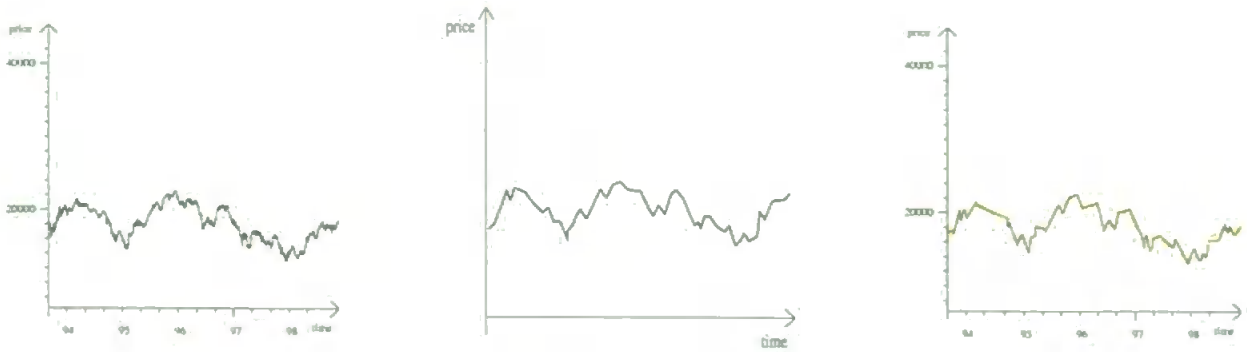


Figure 5.2: **Left**: The original JAPDOW stock index time-series. **Center**: A manual representation of the JAPDOW as obtained by the author. **Right**: A 50 linear segments JAPDOW stock index representation (93% data compression achieved).

In order to make the search and matching process computationally tractable it is desirable to operate on a reduced representation of the time series. In doing so it is desirable to retain important aspects of the data while eliminating uninformative, or noisy, data. There are many simple ways of approximating the data, however they require the selection of fixed parameters, eg. the selection of a window size in a moving average calculation. Since we wish to explore the hypothesis that the rate at which important events occur

in financial time series varies, and that the changes may be occurring more rapidly now than in the past, such a fixed approximation window is not suitable. An algorithm which supports a variable sampling of the time series and achieves high compression rates with minimal information loss, is that proposed by Keogh [92]. The key idea of this approach is to perform a local merging of adjacent segments until the resulting increase in residual error exceeds some threshold. The process results in a set of linear segments of varying length which approximate the original time series, as is shown in figure 5.2.
In effect we have a variable length sampling of the original series, where the sampling intervals are determined by the data and the resulting segmentation shows the times at which the trend in the market changes significantly.

5.2 Dynamic Time Warping (DTW)

Generally, DTW, [14], [82], [151] is a technique for comparing time series patterns which may have different lengths. The DTW algorithm, via dynamic programming, expands and/or compresses the patterns in time in order to define a minimum distance measure between them. Here, DTW is used as part of a metric which calculates the distance between parts of a time series with different length. To construct the DTW matrix Γ, [14], for two univariate subsequences T and M with lengths t and m respectively, the cumulative distances between corresponding subsequent time series elements (determined by individual segments and calculated via dynamic programming) is included in each cell (i, j)[1] of the matrix. DTW then outputs a minimum distance D_{DTW} (equation 5.1) and an optimum path $P(k)$ (see figure 5.3), of size K , known as the warping path, where $max(t, m) \leq K \leq t + m$.

$$D_{DTW}(T, M) = min_w\{\frac{\sum_{k=1}^{K} P(k)w_k}{\sum_1^K w_k}\} \tag{5.1}$$

and

$$P(k) = \{p(1), p(2), \cdots, p(K)\}. \tag{5.2}$$

[1] (i, j) denote the time index of the segments T, M

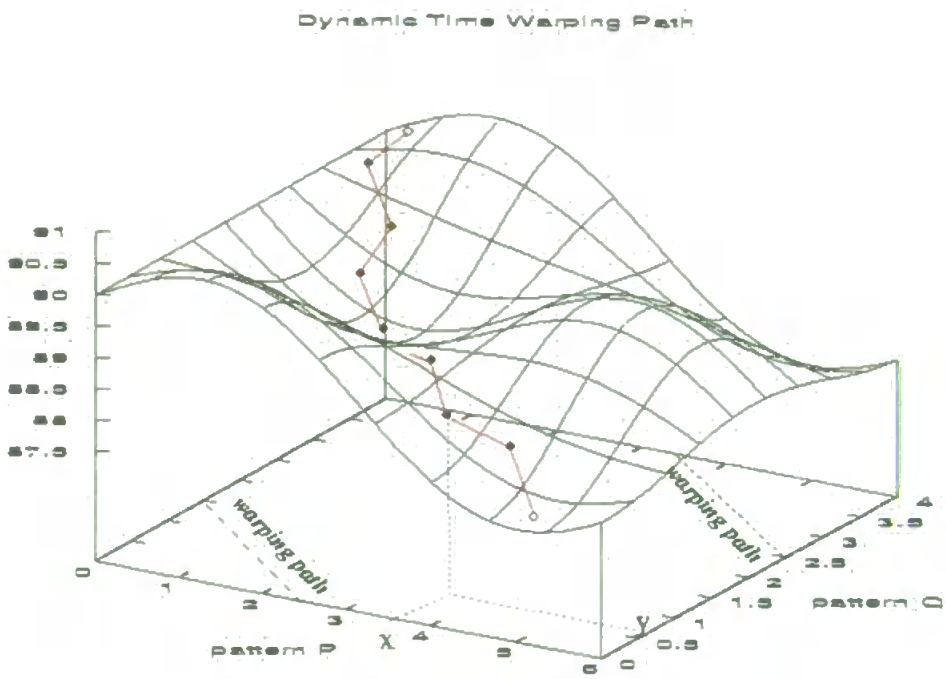


Figure 5.3: A warping path as calculated via dynamic programming using the Dynamic Time Warping algorithm.

w_k, $k = 1 \cdots K$ (equation 5.1) is a sequence of non-negative weights which summate to 1.

The optimisation/minimization problem of equation 5.1 is equivalent to finding the shortest path in the $t \times m$ DTW matrix subject to a number of constraints based on physical consideration, intuition and computational efficiency. These are outlined as follows [152], [14]:

- *Monotonicity*: The points must follow a monotonic trend with respect to time, $i_{k-1} \leq i_k$ and $j_{k-1} \leq j_k$, where i_k, j_k are the coordinates of the $P(k)$ component of the warping path. Furthermore, the warping path is constrained to fall within a designed warping window, i.e. $|i_k - j_k| \leq \omega$, where $\omega > 0$ defines the window width.

- *Start-Endpoint conditions*: This imposes the alignment of the start/end points of both patterns to be matched. In other words, this means that the warping path must satisfy the equations: $p(1) = \gamma(1,1)$ and $p(K) = \gamma(t,m)$. Other variants of the start-end point conditions can be found in [122].

- *Local continuity*: The steps in the matrix are confined to successive neighbouring points, $i_k \leq i_{k-1} + 1$ and $j_k \leq j_{k-1} + 1$.

Following the principles of dynamic programming, the following recursive formulation (equation 5.3) is used to construct the DTW matrix. The cumulative distance $\gamma(i,j)$ is given by the sum of the distance between the corresponding subsequent time series elements determined by individual segments and the minimum of the cumulative distances of the predecessor points.

$$\gamma(i,j) = d(i,j) + min \begin{cases} \gamma(i-1,j) \\ \gamma(i-1,j-1) \\ \gamma(i,j-1) \end{cases} \qquad i = 1,\cdots,t \; and \; j = 1,\cdots,m \; . \quad (5.3)$$

In this section our purpose was to briefly describe the DTW algorithm and the way it can be used to match parts of univariate time series of different length. More detailed studies on the DTW algorithm can be found in KRUSKALL et al., (1983), [99], RABINER et al., (1993), [144] and BERNDT et al., (1994), [14].

5.3 Matching-Prediction-Trading

In this section, we analyse the way that pattern matching is performed on segmented time series. The key idea is to define a similarity measure based on the compound feature information that derives from each individual segment of the query pattern P^Q. We then introduce the notion of Minimum Embedding Segment Dimension (MESD). MESD is used as evidence for non-linear deterministic behaviour of time series underlying dynamics and as an approach to assess the number of successive segments which compose the "current" pattern (see 4.2) to be matched. As in the previous chapter, we perform matching for

forecasting and trading. The innovative aspects of the matching forecasting and trading algorithm are systematically presented here.

5.3.1 Minimum Embedding Segment Dimension (MESD)

The question that many researchers ask themselves when observing financial time series, is whether the dynamics of the data generating process derive from deterministic chaos. To answer this question suitable mathematical theories have been developed first by TAKENS, (1981), [168] and later by SAUER et al., (1991), [157] and OTT et al., (1994) [128]. In these papers different types of embedding dimension calculation methods are discussed. Known as *time-delay embedding*, this theory has also been investigated by many researchers recently[2]. Among others, CECEN et al., (1996), [34], LISI et al., (1997), [103], CAO et al., (1998), [31] and SOOFI et al., (1999), [163] introduced different variants of calculating the optimal embedding dimension of financial time series. By studying their experiments on a set of exchange rate series, it can be seen that the results on the embedding dimensions (ED), that they claim, strongly depend on fixed parameters, like the constant sampling rate of the data. CAO et al., (1999), [33] for example, reported much higher EDs on daily exchange rates than LISI et al., [103], who used monthly data. Here, we calculate the ED of segmented stock indices. We call it Minimum Embedding Segment Dimension (MESD) and it is measured in number of segments, [10]. This overcomes the problem of constant sampling rate and allows the search for attractors in the data, which derive from linear segments of different length. Finally, the MESD calculations are faster than its ancestors.

The Method Consider a time series $x_1, x_2, \cdots, x_N$ which has been divided into M segments ($M < N$), using the method given in 5.1. The resulting segmented series is:

$$S = \{\vec{s_0}, \vec{s_1}, \vec{s_2}, \cdots, \vec{s_i}, \cdots, \vec{s_M}\} \quad , \tag{5.4}$$

[2]Similar research has been undertaken by BLASCO et al (1996), [18], in his effort to identify *long memory* in the Spanish stock market.

where each segment $\vec{s_i}$, corresponds to a segment subset, s_i^* of successive time series values:

$$s_i^* = \{x_j, \cdots, x_{j+k_i}\} \quad . \tag{5.5}$$

k_i is the time duration of each segment $\vec{s_i}$. A time-delay vector z of the segmented time series (equation 5.4) can be written as follows:

$$z_i(d) = \{\vec{s_i}, \vec{s_{i+1}}, \cdots, \vec{s_{i+(d-1)}}\}, \quad i = 0, 1, 2, \cdots, M - (d-1) \quad , \tag{5.6}$$

where d is an integer number considered as the systems embedding dimension. For simplicity, the time-delay parameter has been chosen to be 1. Following the notation in [91], we define in equation 5.7, β as the ratio of the distances of the time-delay segment vectors z_i with their corresponding nearest neighbours, when moving from embedding dimension d to $d+1$. This is illustrated as follows:

$$\beta(i, d) = \frac{\|z_i(d+1) - z_{NN_i}(d+1)\|}{\|z_i(d) - z_{NN_i}(d)\|}, \quad i = 0, 1, 2, \cdots, M - d \quad . \tag{5.7}$$

$\| \cdot \|$ is a distance norm defined as the standard deviation (std) of the distances of the corresponding time series parts indicated by individual segments. This derives from the fact that similar vectors have point distances characterised by low standard deviation. Time series parts are compared using the DTW distance metric. So:

$$\|z_x(d) - z_y(d)\| = std(D_{DTW}(s^{*x}_1, s^{*y}_1), D_{DTW}(s^{*x}_2, s^{*y}_2), \cdots, D_{DTW}(s^{*x}_d, s^{*y}_d)) \quad . \tag{5.8}$$

D_{DTW} is the minimum distance metric defined in equation 5.1. NN_i is an index which indicates the position of nearest neighbour of the time-delay vector z, on the segmented time series. NN_i is the same in both the numerator and denominator of equation 5.7. According to the embedding theorems of [168] and [157], d is chosen to be the system's ED when two time-delay vector points mapped in the d-dimensional reconstructed space, will remain mapped in the $d+1$-dimensional space. In effect the β-ratio defined in equation 5.7 will have a value close to one for a suitable minimum embedding segment dimension

	UK	FRANCE	GERMANY	SPAIN	ITALY	GREECE
MESD	24	21	22	16	17	16

Table 5.1: The Minimum Embedding Segment Dimensions of a set of European stock indices, as given by DATASTREAM Inc. The MESD is measured in number of segments.

d. However, because β might be different for every segment's time-delay vector z_i, we adopt the quantity defined by Cao, (1998), [31] to overcome this problem.

$$E(d) = \frac{1}{N-d} \sum_{i=1}^{N-d} \beta(i,d) \tag{5.9}$$

$E(d)$ in equation 5.9 solely depends on the value of dimension d (time delay has been set to one). To investigate the variations of $E(d)$, Cao introduces the fraction $W(d) = E(d+1)/E(d)$. The value of d for which the fraction $W(d)$ stops changing substantially, indicates that the MESD of the underlying system is $d+1$. However, the choice of the threshold that identifies the stopping point is not necessarily clear. Here, its validity is testing through the overall prediction and trading performance of the system. By applying the above described algorithm on segmented representations of a set of stock indices, we have found indications of non-linear deterministic behaviour. The MESDs on the whole data set is given in table 5.1. In figure 5.4 some example curves, as obtained from the MESD algorithm, show the embedding segment dimension values.

5.3.2 Multiple Feature Sets (MFS)

The purpose of the matching algorithm is to find past occurrences similar to the "current" time series situation, described by the query pattern. This time the query pattern is depicted with a sequence of segments, whose number is assessed by the MESD method. The detection of such query patterns in the history of financial time series requires a flexible matching process which can retrieve the patterns' time dependencies. Consider for example, the upwards trend in the JAPDOW stock index starting at early 1993, the one starting at 1995 and the most recent one, as depicted in figure 5.5.

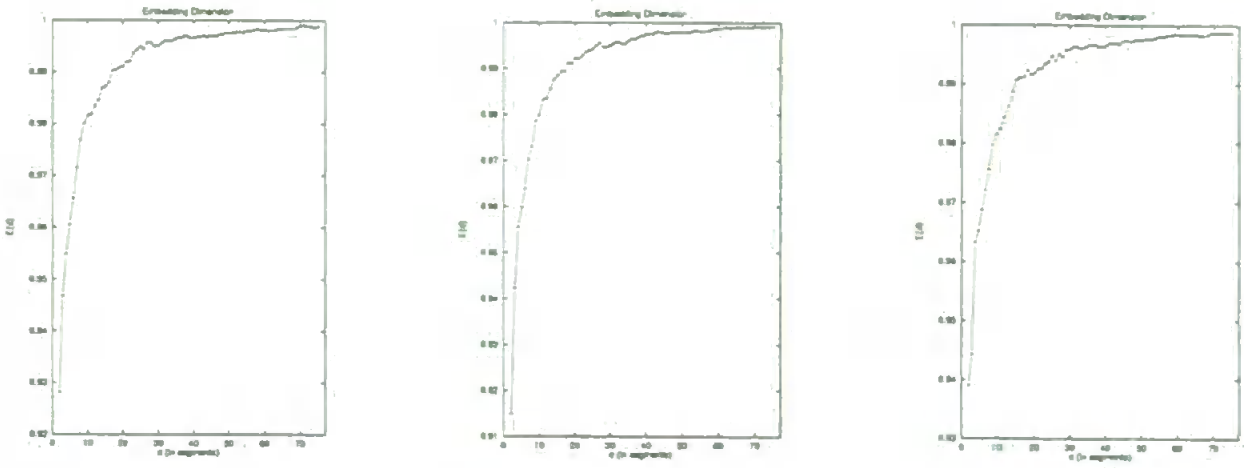


Figure 5.4: Three paradigms of the ESD curves on a set of DATASTREAM market indices. **Left:** FRANCE. **Center:** GERMANY. **Right:** UK. For each curve, the values of d for which $E(d)$ stops changing substantially correspond to the minimum embedding dimension.

All rises have similar shape but occur in different time scales. Although, economists are very good at visually classifying such patterns, to achieve that computationally is a hard task because of the difficulties of matching patterns with some notion of time and shape fuzziness. The Multiple Feature Sets (MFS) matching algorithm we propose, addresses this issue. These feature sets incorporate DTW (section 5.2) to compare similar segment subsets[3] of different duration, the linear segment's 1st derivative function, i.e.

[3] A segment subset is the time series part corresponding to an individual segment

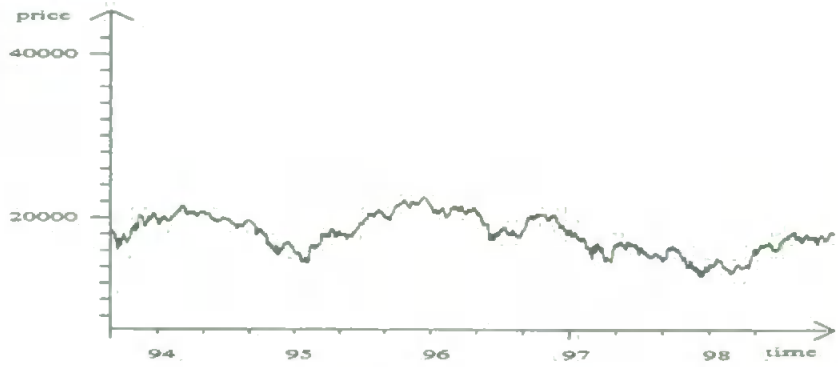


Figure 5.5: The JAPDOW index from late 1993 until 1999. Similar trends appear in the index but with different durations.

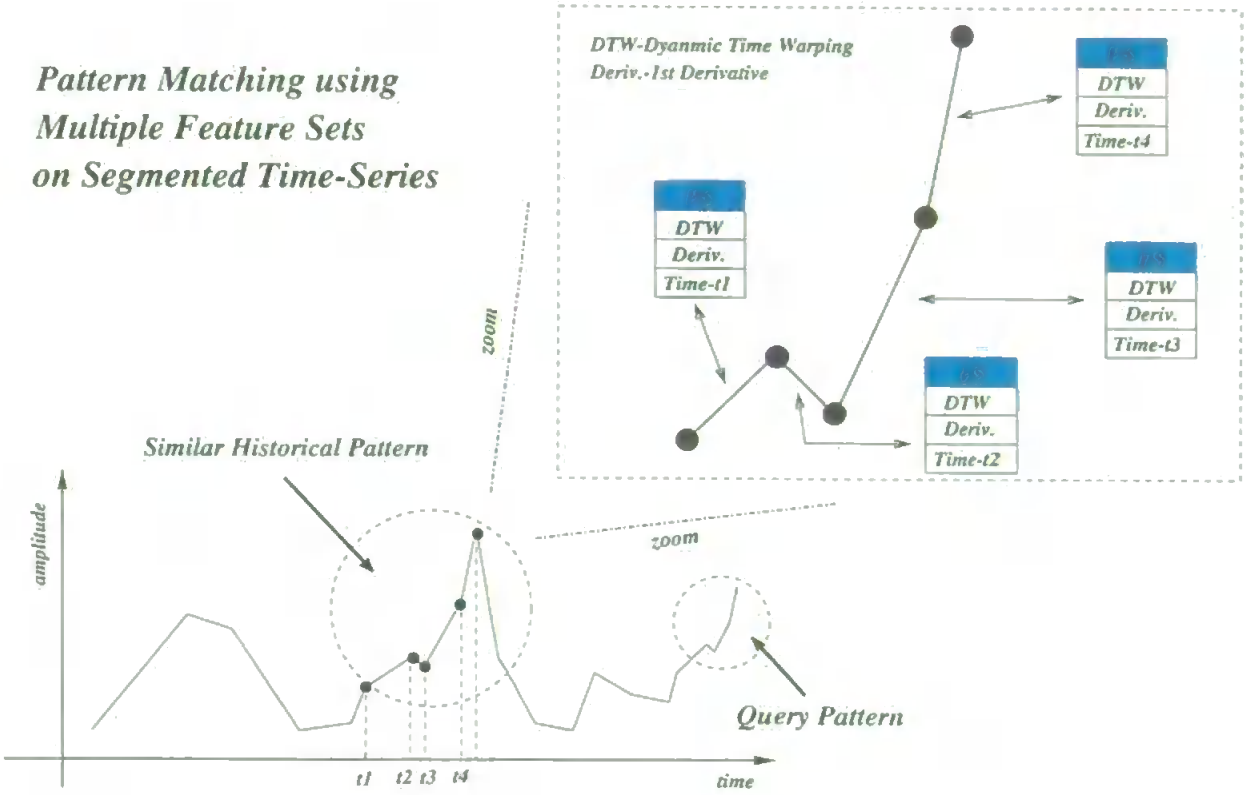


Figure 5.6: Pattern Matching using Multiple Feature Sets on Segmented Time-Series.

the slope for linear segments, to include pattern orientation information, and time labels to distinguish more recent patterns from older ones. In figure 5.6, we depict the way that each MFS is attached to each of the corresponding segments.

Dynamic Time Warping Distance Measure: As explained in section 5.2, DTW effectively maps time series with different number of points in a satisfactory way that can be used to extract a reliable distance measure between them. Because of the optimisation component of DTW, it is advisable to use it on short subsequences. Segment subsets comparison issues are ideal for the application of the DTW algorithm. While sequentially scanning the segmented time series $S = \{\vec{s_0}, \vec{s_1}, \cdots, \vec{s_i}, \cdots, \vec{s_m}\}$ using the query pattern $P^Q = \{\vec{s^Q_{m-d}}, \cdots, \vec{s^Q_{m-k}}, \cdots, \vec{s^Q_m}\}$, $k = 0, 1, \cdots, d$,[4] the distance between P^Q and $P_i = \{\vec{s_i}, \cdots, \vec{s_{i+k}}, \cdots, \vec{s_{i+d}}\}$, $i = 0, 1, \cdots, m-(d-1)$, is defined as the average of the distances

[4] d is the minimum embedding dimension of S.

measured with the DTW metric $D_{DTW}(s^{*Q}_i, s^*{}_i)$ on the individual segment subsets[5]. So:

$$D_G(P^Q, P) = \frac{1}{d}\sum_{i=1}^{d} D_{DTW}(s^{*Q}_{m-d+i}, s^*{}_i) \tag{5.10}$$

Note that in the DTW distance calculation (equation 5.10) the segment subsets as defined in equation 5.5 are used. D_G is the first feature used to find the best matched position for the query pattern P^Q.

Orientation Information: The second feature which enriches the multiple feature sets and therefore the whole pattern matching process is the slope of the segment in process, which can be seen as the tan^{-1} of the 1st derivative of the segment's linear function. This feature actually distinguishes a part of the time series corresponding to a segment, from another with the same range of magnitude but different orientation. This is the case where the usual distance measures fail.

Time Labels: Finally, a time label feature is included in the multiple feature set. Matching patterns are compared with the value of their similarity. To overcome the problem of more than one pattern having the same similarity, we make use of the time labels and select the most recent one. The way that each multiple feature set hangs on each corresponding segment is shown in figure 5.6.

The feature set, described above, is integrated in the following similarity function D_G:

$$D_G(P^Q, P) = \frac{1}{d}\sum_{i=1}^{d} \frac{w_{DTW} D_{DTW}(s^{*Q}_i, s^*{}_i) + w_{SLOPE} D_{SLOPE}(\vec{s_i^Q}, \vec{s_i})}{d} \quad , \tag{5.11}$$

where $D_{DTW}(s^{*Q}_i, s^*{}_i)$ is the minimum distance norm calculation, according to the optimum path found via dynamic programming using the Dynamic Time Warping algorithm (see 5.2 and [99, 14]), $D_{SLOPE}(\vec{s_i^Q}, \vec{s_i})$ is the squared difference of the slopes of the segments. $\vec{s_i}$ and $s^*{}_i$ refer to the segment and its corresponding time series subset at position i respectively.

[5]The start and the end point of each time series subset $\vec{s_i}$, is identified by the corresponding segment s_i. A way of directly applying DTW on segmented time series has been proposed in [93]

In equation 5.11, the distance measures for each feature are computed separately. Each constituent feature similarity is then added with certain normalized weighting factors. Satisfactory weighting can be found experimentally. Here, we have selected $w_{DTW} = 0.6$ and $w_{SLOPE} = 0.4$ ($w_{DTW} + w_{SLOPE} = 1$). For this weighting set, desired matches are achieved experimentally when similarity is greater than 60%

5.3.3 Trend Prediction with Time Considerations / Trading

Trend Prediction: Once the "current" situation of the market (i.e. the query pattern P^Q) has been matched to a certain position, i.e. it is geometrically similar to what happened sometime ago in the time series, we hypothesise that the local reaction of the market after the historical matched occurrence will indicate some shape and magnitude information about the market's future activity. Economic or technical analysis of that information might also be possible at that stage. This kind of prediction system thinking becomes more valuable, because of affinities in the market's behaviour, derived solely from well established economic market interpretation theory, from broadly used technical analysis tools or even from the basic rule of *demand and supply.* Imponderable factors, such as political guidance or intervention and insider trading, are not considered here. Figure 5.7 depicts a graphical explanation of the above prediction hypothesis. Similar patterns captured with the MFS matching algorithm are shown within the dashed circles. Note that pattern P is stretched in time and amplitude relative to the query pattern P^Q. As a consequence of this, an analogous shrunken version of the segments following pattern P (bold lines outside the historical pattern P circle) is taken to be the predictive indication for the time series future activity (bold line at the end of the series). Technically speaking, each trend prediction is normalized in duration, according to the ratio of the durations of patterns P and Q. Moreover, analysis, based again on temporal ratios, that verifies whether the last univariate time series point is the end or part of the current trend, is performed as described below. So, the main issues here are:

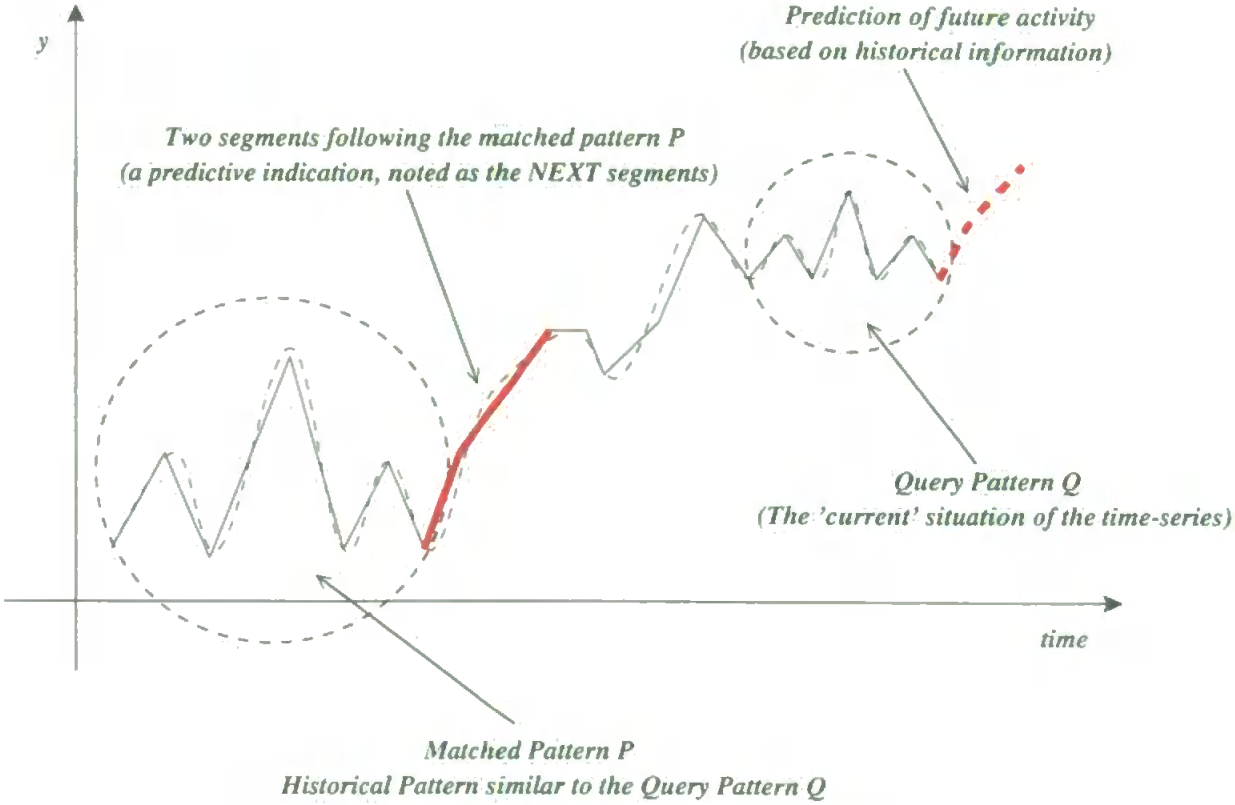


Figure 5.7: How trend prediction can be made using multiple feature set pattern matching.

1. **Time Normalization**:

   Let's assume that the matched pattern occurred for time Δt_P which is the sum of the duration of each segment:

$$\Delta t_P = \Delta t_{P_1} + \Delta t_{P_2} + \cdots + \Delta t_{P_m} \tag{5.12}$$

   where m is the number of segments of the pattern. Using the same notation, the query pattern has time length Δt_Q. If $\Delta t'_{P_1}$ stands for the duration of the segment that follows the matched pattern, then the model indicates that the predicted linear segments will last for time:

$$\Delta t'_{Q_1} = \Delta t'_{P_1} * \frac{\Delta t_Q}{\Delta t_P} \tag{5.13}$$

   For more than one step ahead prediction we form analogous calculations.

2. **Edge Point Verification**:

   A weakness of this prediction approach as described up to now is that it assumes that the last segment of the query pattern is a "*stop*" point i.e. in the next moment a new segment is about to appear. That might not be true and most possibly the current segment will continue following its own trend for some time in the future before changing direction. For that purpose we check the relation of the ratio r_{last} of the time length of the last segment of the pattern Q and the time length of the corresponding segment of the matched pattern P ($r_{last} = \frac{\Delta t_{Q_{last}}}{\Delta t_{P_{last}}}$) against the patterns length ratio r ($r = \frac{\Delta t_Q}{\Delta t_P}$). If $r_{last} \geq r$, we take the last segment to have finished its activity. In the opposite case the last segment will continue for time $t = \Delta t_{P_{last}} * \frac{\Delta t_Q}{\Delta t_P} - \Delta t_{Q_{last}}$.

3. **Confidence Measures**:

   - Matching Confidence - The similarity value (%) measured between the query pattern, P^Q, and its best match P.
   - Prediction Confidence - A measure (%) of how well the k segments (lines) that follow the best match, P, fit the actual time series.

   We use these confidence measures to check the improvement on matching and prediction accuracy and to drive a selective trading strategy.

Trading: Basically, a trading system is a hybrid of two main modules. A prediction module, which optimises its performance to output accurate predictions and a trading module, which takes trading positions (*buy-sell-hold*) according to a set of trading rules and to the predictions originated from the prediction module. Other more sophisticated types of trading systems have been proposed [119, 184, 120, 171]. Here, we base trading on the performed segment predictions using simple trend following trading rules. If no change in the trend from the previous segment is predicted, the current trading position is maintained and unnecessary brokerage costs are avoided. The trading rule, we employ,

is summarised as follows:

$$if \quad SL(s_{t-1}) > 0 \;\&\&\; SL(s_t) > 0 \;\&\&\; SL(s_{t+1}) < 0 \quad sell - go\ short \quad , \tag{5.14}$$

$$if \quad SL(s_{t-1}) < 0 \;\&\&\; SL(s_t) < 0 \;\&\&\; SL(s_{t+1}) > 0 \quad buy - go\ long \quad . \tag{5.15}$$

$SL(s_{t-1}), SL(s_t), SL(s_{t+1})$ are the slopes of the previous, current and predicted segments respectively[6]. Each transaction takes place at time t, guided by the duration of the predicted segment. The trading rules of equation 5.14, 5.15 are further enriched making use of the slope indication for each predicted segment. We force the trading system to buy(sell) more(less) stocks, depending on how aggressive the segment has been forecasted to be. Assuming that α is the slope of the predicted segment and α_G is the slope of the regression line fitted on the whole time series, we set the following trading constraints:

$$if \quad \frac{|\alpha|}{|\alpha_G|} \leq 1, \quad sell/buy\ one\ stock \quad , \tag{5.16}$$

$$if \quad \frac{|\alpha|}{|\alpha_G|} > 1, \quad sell/buy\ \lfloor \frac{\alpha}{\alpha_G} \rfloor\ stocks \quad . \tag{5.17}$$

Brokerage costs are considered to be 1% of the transaction price. Finally, a stop loss criterion is considered for the system. Every time that losses exceed a 10 basis points threshold, open trading positions close in order to prevent extensive losses. The automatic trader then waits for a new trading signal in order to take further action. The trading rules system is activated when the prediction confidence is more than 70%. The profit achieved is accumulated all over the testing interval.

5.4 Evaluation

The evaluation set used to test the system previously described, is composed of six European market indices. To increase objectivity about the generalization properties of the system, we chose to work with three market indices from north European countries and another three from countries bordering the Mediterranean. These are the UK, France,

[6] $SL(s_{t+1})$ is replaced with $SL(s_{t+a})$ for a-segs ahead predictions.

Germany, Spain, Italy and Greece[7]. Data have been downloaded from the DATASTREAM[8] database, as it is processed by the DATASTREAM team. The time series range from 01.01.1990 to 26.04.2000, in daily representations (2693 working days). The compression achieved with the segmentation algorithm is on average 84%. The results we present here have been obtained from the last 40% of the time series points.

5.4.1 Non-linear Deterministic Behaviour

All the European indices employed in this study, reveal non-linear deterministic behaviour, according to the minimum embedding segment dimension (MESD) measures. In figure 5.8, we depict the variation of $E(d)$ (equation 5.9) while d varies between 2 and 50. It can be seen there that $E(d)$ stops changing substantially after a value D_{min}. D_{min} corresponds to the selected MESD. This behaviour may be some evidence that the data in process are not generated by purely random processes. Purely random generated series do not follow patterns such as those depicted in figure 5.8. However, because high dimensional systems may be in practice indistinguishable from stochastic systems, the possibility that the market indices are generated from non-linear stochastic processes cannot be excluded. As shown in table 5.1, the ESD for the European indices ranges between 25 and 35 segments.

As mentioned earlier, MESD is measured in number of segments. An attempt to transform MESD in number of days ($\sum_{i=1}^{MESD}(s_i\ duration)$) establishes the claim made by Cao, [33], that the minimum embedding dimension (MED) of financial data is usually high. According to our measures the MED of the European data set varies between 50 and 70 days. High MED are computationally expensive to calculate. Using segmented representations and segmented embedding dimension calculations, reduces the numbers by at least a factor of two. For the tests undertaken here the calculations have been reduced on average by a factor of ten. The fact that the ESD calculations are reduced

[7]We refer to the following market indices: UK-DS-MK, FRANCE-DS-MK, GERMANY-DS-MK, SPAIN-DS-MK, ITALY-DS-MK, GREECE-DS-MK

[8]Datastream International Limited, Monmouth House, 58-64 City Road, London EC1Y 2AL

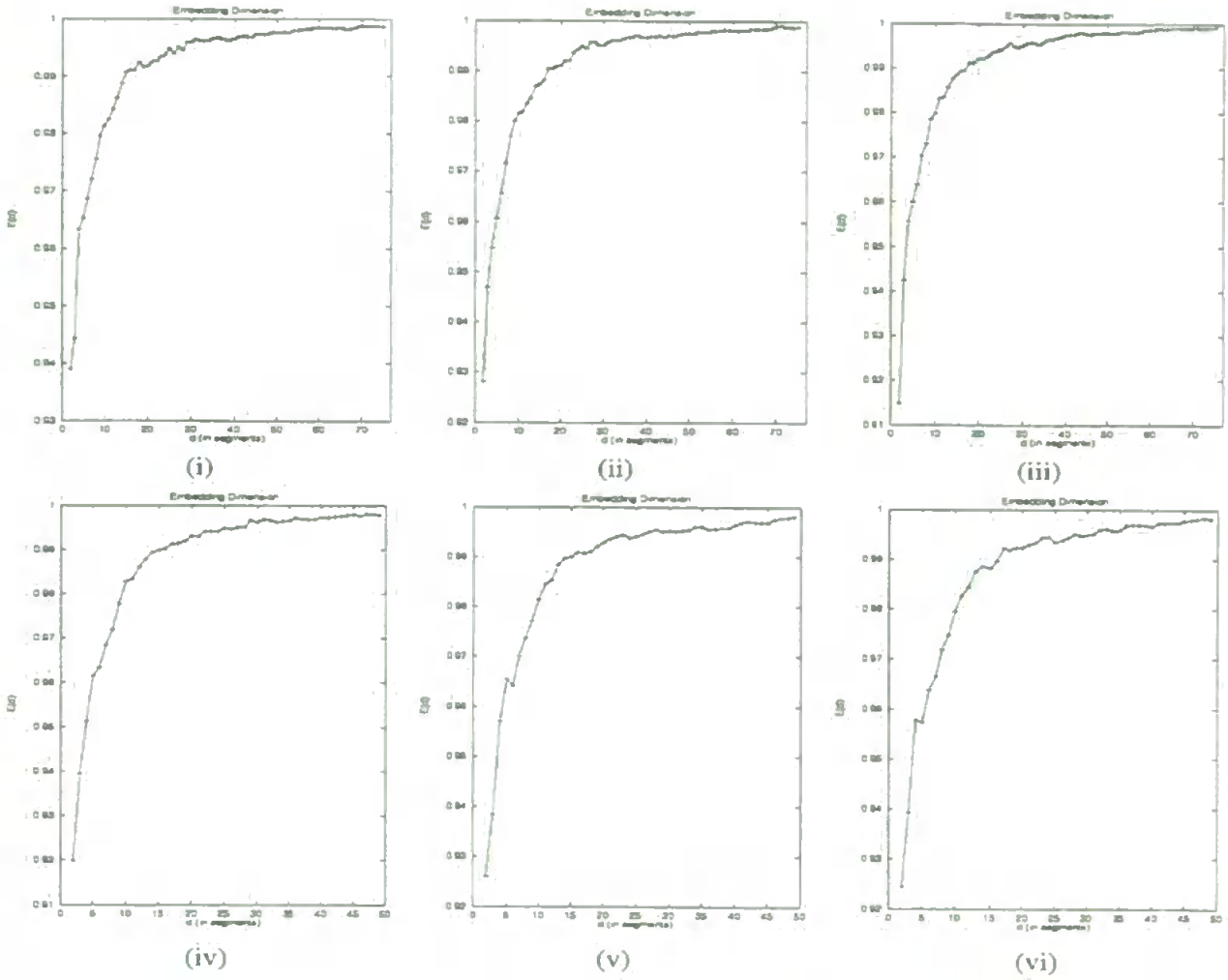


Figure 5.8: Identifying the Minimum Embedding Segment Dimension (MESD) of a set of six European indices. The curves depicted above are $E(d)$ values for different embedding dimensions, d (see equation 5.9). The d value for which $E(d)$ stops changing substantially is taken to be the minimum embedding dimension of the system. **(i)**: The UK-DS-MK index case. **(ii)**: The FRANCE index case. **(iii)**: The GERMANY index case. **(iv)**: The SPAIN index case. **(v)**: The ITALY index case. **(vi)**: The GREECE index case.

by at least a factor of two, makes our pattern matching, forecasting and trading system computationally faster than its predecessor.

5.4.2 Evidence for speed-up

In table 5.2 we give some evidence in support of the hypothesis, that mentioned previously, that there is a speedup in the market's interaction possibly due to external factors, such as the technology and the telecommunications expansion. As shown in table 5.2 for different query pattern lengths, measured in number of segments, the percentage of the best historical matches with duration longer than the one of the current query pattern exceeds 70% for all the European stock indices. The matching test has been performed on 40% of the data, which corresponds to 1077 working days.

Market Index	3	4	5	6	7	8	9	10	Average
UK	86%	86%	88%	87%	88%	83%	85%	86%	86.1%
FRANCE	86%	84%	82%	83%	86%	87%	82%	81%	83.9%
GERMANY	80%	75%	80%	83%	83%	81%	86%	86%	81.8%
SPAIN	86%	75%	72%	71%	74%	78%	75%	71%	75.3%
ITALY	89%	86%	86%	89%	87%	87%	83%	85%	86.5%
GREECE	85%	80%	81%	82%	86%	80%	84%	89%	83.4%
Total Average									82.8%

Table 5.2: More than 70% of the historical matches are longer in duration than the current query pattern. The hypothesis has been tested on 40% of each financial data series (six European DATASTREAM market indices between 01.01.1990 - 26.04.2000). Each column represents the query pattern length in number of segments.

Furthermore, in figure 5.9 some visual evidence to support this hypothesis is given. In this figure, the distributions of the durations (lengths in number of points) of the segments that represent the European market indices are depicted. It can clearly be seen, that the segment durations tend to decline as the series evolves through time. According to these results, it becomes critical for the forecasting performance of a pattern matching system to

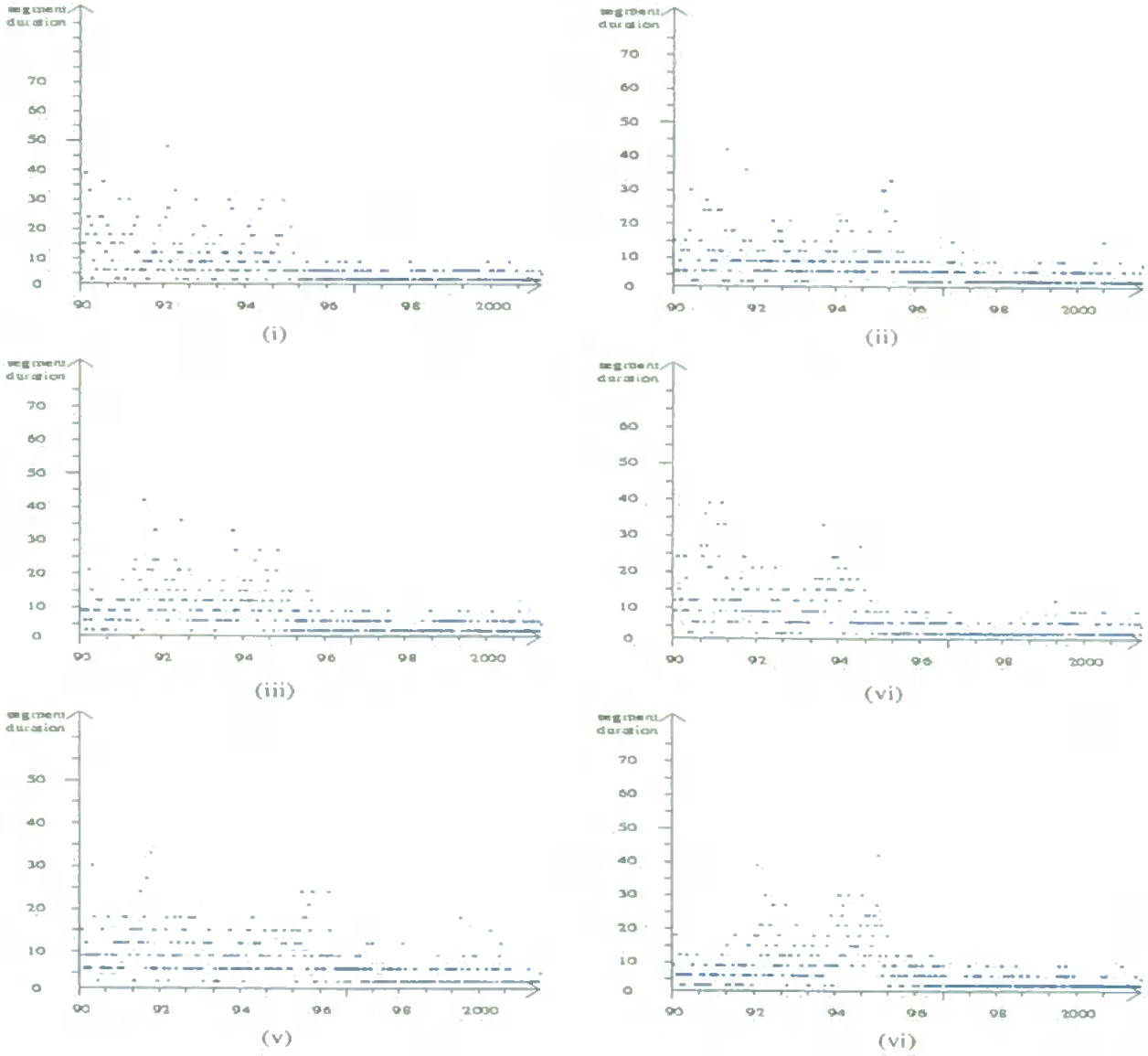


Figure 5.9: Distribution of segment duration(length) through time for six European market indices. A clear concentration of long segment lengths can be seen in the early stages of all series. **(i)**: The UK index case. **(ii)**: The FRANCE index case. **(iii)**: The GERMANY index case. **(iv)**: The SPAIN index case. **(v)**: The ITALY index case. **(vi)**: The GREECE index case.

be able to capture this dynamic time evolving change. This is exactly what the proposed segmented pattern matching algorithm attempts to accomplish by using dynamic time warping. The distributions in this figure correspond to the period from 01.01.1990 to 26.04.2000 for six European market indices. Similar results have been obtained with

most of the known stock indices and some stock assets.

5.4.3 Prediction Results

Tables 5.3 and 5.4 show the accuracy of the system in predicting one segment ahead. The query pattern length is fixed using the MESD indications for each of the time series. Each prediction is accompanied by average matching and prediction confidence measures.

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
DA	58.06	63.33	58.06	55.17	53.00	**57.52**	**58.41**
AMC	66.71	63.85	64.21	62.05	64.39	**64.24**	**60.60**
APC	58.30	61.34	63.54	63.88	60.58	**61.53**	**58.69**
UK-DS-MK							
DA	64.29	66.67	62.50	48.15	75.76	**63.47**	**60.86**
AMC	67.88	69.00	64.61	63.02	68.12	**66.53**	**61.55**
APC	62.72	64.41	66.05	60.42	63.05	**63.33**	**59.38**
FRANCE-DS-MK							
DA	62.96	70.97	64.52	44.44	59.14	**60.41**	**57.45**
AMC	67.95	65.03	62.83	60.29	71.54	**65.53**	**60.91**
APC	66.13	65.46	62.30	60.59	65.57	**64.01**	**58.96**
GERMANY-DS-MK							

Table 5.3: Testing the one-segment ahead forecasting directional ability (%) of the segmented pattern matching system on the European DATASTREAM market indices set (UK, FRANCE, GERMANY). DA, AMC, APC stand for Directional Ability, Average Matching Confidence and Average Prediction Confidence respectively.

This prediction scheme has been tested on several 2-year and 10-year periods of six European DATASTREAM market indices. The minimum and the maximum correct direction prediction percentages for all the 2-year index periods are 44.44% and 75.76% respectively. These figures translate to an average directional ability of 59.31% for the

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
DA	59.37	57.45	62.82	60.49	55.73	**59.17**	**59.55**
AMC	70.29	64.36	66.21	63.68	64.15	**65.74**	**59.67**
APC	63.15	64.49	64.01	66.12	64.30	**64.41**	**61.01**
SPAIN-DS-MK							
DA	64.29	58.65	66.00	48.15	60.00	**59.42**	**64.68**
AMC	66.01	61.49	67.77	64.03	66.11	**65.08**	**63.84**
APC	62.34	61.53	60.82	68.51	60.31	**62.70**	**61.36**
ITALY-DS-MK							
DA	51.43	56.72	56.17	52.15	62.76	**55.85**	**58.43**
AMC	64.52	64.13	66.57	63.45	62.80	**64.29**	**58.62**
APC	64.48	64.01	68.04	64.15	61.57	**64.45**	**56.63**
GREECE-DS-MK							

Table 5.4: Testing the one-segment ahead forecasting directional ability (%) of the segmented pattern matching system on the European DATASTREAM market indices set (SPAIN , ITALY GREECE). DA, AMC, APC stand for Directional Ability, Average Matching Confidence and Average Prediction Confidence respectively.

2-year long market indices. The corresponding value for the 10-year period is 59.90%. In addition, not only does the prediction performance of the segmented pattern matching scheme reach an average of 60% in terms of correct direction prediction but it also exhibits robustness in terms of the duration of the series that it has been tested on. The system seems to perform similarly for shorter and for longer time periods. This effect has not been recorded for the pattern matching scheme of chapter 4. Finally, the matching confidence is over 60% for most of the time. This indicates the reliability of the similarity function and the matching algorithm.

5.4.4 Trading

Following the strategy described in section 5.3.3, we trade on each of the European DATASTEAM indices independently, starting from 01.08.1995 until the 26.04.2000. This corresponds to 1236 working days. The strategy followed, extracts on average 138 trading actions per index. This number has been achieved after applying the *stop loss* criterion (section 5.3.3) on the 10-year long market indices. These figures are correspondingly different for the 2-year periods. The average profits gained for each index are represented in tables 5.5 and 5.6.

	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Average	'90-2000	B/H
CP	14.47	9.43	19.69	15.60	7.52	**13.34**	**8.61**	13.49
SR	2.60	1.03	2.11	1.31	1.09	**1.63**	**1.04**	
NoT	31	30	31	29	32	**31**	**136**	
UK-DS-MK								
CP	17.74	10.23	18.33	20.02	12.41	**15.75**	**25.33**	12.76
SR	2.13	1.48	2.69	2.87	1.95	**2.22**	**2.66**	
NoT	28	33	32	27	33	**31**	**140**	
FRANCE-DS-MK								
CP	19.78	12.77	13.16	-9.20	19.22	**11.15**	**19.14**	4.41
SR	2.35	1.41	1.87	-0.99	2.46	**1.42**	**2.37**	
NoT	27	31	31	27	28	**29**	**137**	
GERMANY-DS-MK								

Table 5.5: Testing the profitability (basis points) of the segmented pattern matching system on the European DATASTREAM market indices set (The UK, FRANCE and GERMANY case). CP, SR, NT stand for Cumulative Profit, Sharpe Ratio and Number of Transactions respectively. B/H corresponds to the *Buy and Hold* (B/H) trading strategy.

For all market indices the average profit gained for the 2 and 10 - year periods is 15.43 and 16.16 basis points respectively. Prediction confidence greater than 60% has

	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Average	'90-2000	B/H
CP	26.38	-1.06	7.76	33.65	25.69	**18.48**	**6.07**	2.23
SR	6.47	-0.20	1.89	7.33	7.01	**4.50**	**1.46**	
NoT	28	30	28	26	28	**28**	**138**	
SPAIN-DS-MK								
CP	9.16	14.37	23.44	29.04	32.55	**21.71**	**13.69**	14.64
SR	1.17	1.93	3.44	4.52	5.19	**3.25**	**1.87**	
NoT	28	37	30	27	28	**30**	**139**	
ITALY-DS-MK								
CP	18.68	12.05	-9.20	17.75	21.53	**12.16**	**24.09**	12.90
SR	2.54	2.04	-1.42	2.39	3.96	**1.90**	**4.11**	
NoT	28	29	29	26	29	**28**	**140**	
GREECE-DS-MK								

Table 5.6: Testing the profitability (basis points) of the segmented pattern matching system on the European DATASTREAM market indices set (The SPAIN, ITALY and GREECE case). CP, SR, NT stand for Cumulative Profit, Sharpe Ratio and Number of Transactions respectively.

been taken as the constraint for the automatic trader to enter the market. In tables 5.5 and 5.6, these figures together with Sharpe Ratio (SR) risk measures are depicted analytically. In this table the average profit numbers that are generated by the "*Buy & Hold*" trading strategy adjusted to the number of transactions are given too.
In tables 5.7 and 5.8, similar trading performance figures are presented after enhancing the trading strategy with the slope rules given by the equations 5.16 and 5.17. Improvement in terms of profit can be seen in this table. However, the number of transactions drops substantially when applying slope selective trading. Therefore, the increase in the trading performance is amplified by the decrease of the overall transactions costs payed.

Representative mean cumulative curves over the 1236 working days periods are depicted in figure 5.10. These curves are characterised by upward trends.

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000	B/H
CP	15.93	10.06	16.49	20.23	10.05	**14.55**	**10.19**	13.49
SR	2.80	1.27	2.36	2.77	2.56	**2.35**	**1.17**	
NoT	9	10	17	11	9	**11**	**33**	
UK-DS-MK								
CP	16.91	17.61	16.22	20.12	12.03	**16.58**	**26.18**	12.76
SR	1.98	2.36	3.83	2.27	2.59	**2.61**	**2.75**	
NoT	16	21	17	8	14	**15**	**40**	
FRANCE-DS-MK								
CP	21.72	14.87	14.16	1.31	24.69	**15.35**	**23.09**	4.41
SR	2.31	1.83	2.01	0.10	2.88	**1.83**	**2.67**	
NoT	15	16	7	9	17	**13**	**34**	
GERMANY-DS-MK								

Table 5.7: Testing the profitability (basis points) of the segmented pattern matching system on the European DATASTREAM market indices set using confidence measures (The UK, FRANCE and GERMANY case). CP, SR, NT stand for Cumulative Profit, Sharpe Ratio, and Number of Transactions respectively. Note that the difference between this table and table 5.5 is that here confidence measures are used for trading as well.

In conclusion, it can be said that: 1) the trading results show that the average profit measures constantly have positive signs; 2) the number of trading transactions indicate that the *automatic trader* acts only for almost 10% of the time applied and this avoids excessive brokerage costs and risky every day trading.

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000	B/H
CP	20.74	6.84	12.25	27.06	26.38	**18.65**	**11.73**	2.23
SR	2.26	1.39	1.52	3.06	2.73	**2.19**	**1.12**	
NoT	16	18	13	9	15	**14**	**34**	
SPAIN-DS-MK								
CP	10.66	13.29	25.67	25.88	28.00	**20.70**	**14.49**	14.64
SR	1.62	1.76	4.15	4.44	3.48	**3.09**	**2.78**	
NoT	12	10	12	11	11	**11**	**48**	
ITALY-DS-MK								
CP	28.80	16.24	-5.28	20.42	24.58	**16.95**	**17.32**	12.90
SR	4.72	3.14	-0.86	3.09	4.05	**2.83**	**2.48**	
NoT	13	8	17	11	13	**12**	**23**	
GREECE-DS-MK								

Table 5.8: Testing the profitability (basis points) of the segmented pattern matching system on the European DATASTREAM market indices set using confidence measures (The SPAIN, ITALY and GREECE case). CP, SR, NT stand for Cumulative Profit, Sharpe Ratio, and Number of Transactions respectively. Note that the difference between this table and table 5.6 is that here confidence measures are used for trading as well.

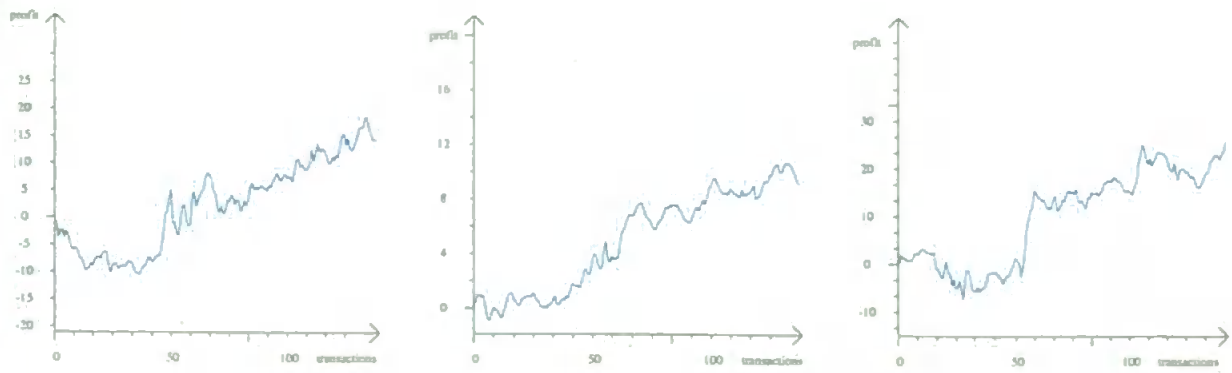


Figure 5.10: The evolution of profit (in basis points) for the GERMANY (left), UK (center) and GREECE (right) datastream market indices starting from 01.08.1995 - 26.04.2000.

5.5 Conclusion

We have shown a way of variable sampling of financial time series based on time series segmentation. Then, a pattern matching system, applied to segmented time series representations, has been described. The main purpose of this system was to detect similar time series patterns, which occur at different time scales, in order to perform trend prediction with time and slope indications. The pattern matching system was based on embedding segment dimension calculations and on dynamic time warping.
The conclusions that have been made after the system was applied to a set of European market indices, can be summarised in the following:

- The segmented time series representations and the pattern matching algorithm have shown that there is a *speed-up* in the evolution of the market indices, that may be due to external factors such as the technology and the telecommunications expansion in finance.

- The use of the forecasting pattern matching system has achieved prediction accuracy which on average exceeded 59%. The trend's slope and duration predictive indications were also satisfactory.

- The transformation of the system's trend predictions into trading actions resulted in average returns which were positive. Furthermore, the trend of the profitability curves over the testing periods of the market indices was also upward. The trading mechanism that has been used for those experiments has acted on average for only 10% of the time. Therefore, unnecessary trading costs have been avoided.

Chapter 6

Exploring regression relationships

"The future isn't what it used to be."

Anonymous,

'Supplied by Joseph Silling'

Until now, our investigations have been concentrated on univariate financial time series analysis with respect to predictability and investment decision making. The multivariate case will be addressed in this chapter. The field of multivariate analysis generally consists of those techniques that consider two or more related variables as a single entity and attempts to produce an overall result taking the relationship among the variables into account. Looking back to time series literature, it is easy to track a vast dominance of univariate time series studies in contrast to the multivariate ones. In principle, univariate scalar time series are on their own sufficient to discover the dynamics of the underlying systems if enough historical data is available. However, in practice the situation may be different. A representative example to illustrate this is the reconstruction of the Lorenz equation, [108], from its z-values. The z-values cannot on their own explain the dynamics of the Lorenz system because they cannot resolve the $x - y$ symmetry of the system. Examples from other application domains also suggest that trivial problem solutions considered in a M-dimensional space cannot be solved in lower $k < M$ dimensions[1].

[1]The XOR classification problem is trivial when seen in two dimensions but impossible to be solved in one. The *Support Vector Machines* is an advanced theory which attempts to offer problem solutions

Therefore, we cannot be sure in practical time series problems, as those discussed here, whether one scalar time series can be sufficient for analysing the dynamics of the system. The use of several different time series might add considerable advantages in predicting noisy economic time series data.

Economic data is usually multidimensional. In this chapter, we first discuss the prediction problem of multivariate time series as a generalization of the univariate procedures, we proposed in chapters 4 and 5. Generally, we operate under the hypothesis that multiple economic data streams, if used effectively for modelling and prediction, can significantly improve predictability and investment decision making. We evaluate this approach on the same set of European stock indices in order to allow a straightforward comparison to the univariate performance given in the previous chapters and illustrate the improvement achieved using more relevant data sources for prediction and trading. In the second part of this chapter, we deal with the concept of co-integration (see ENGLE and GRANGER (1987) [52]) between economic time series. Within the framework of co-integration, which simply tests whether various market indices and/or economic variables reveal common movement in the long run while showing temporary divergences, we propose here a systematic method for exploring co-integration within a temporal framework which reveals local time relations between the time series in process. We call this type of co-integration **time-warped co-integration**. It is our belief that non-synchronous movements, i.e. local time shifts and delays or shrinks and stretches, occur locally in economic time series when seen relative to each other. Therefore, adding local relative time information when looking for co-integration between economic time series may reveal more pragmatic co-movement inferences. The algorithm used to identify those local relative time relations is the Dynamic Time Warping Algorithm (DTW) (see chapter 5). DTW performs a non-linear "*one-to-many*" points mapping of the time series which reveals local relative time delays and shifts between economic time series. *Time-warped co-integration* is proposed as a new equilibrium relationship between economic time series, whose mispricing disequilibrium error correction mechanism is considered for trading arbitrage opportunities.

by increasing the problem's dimensionality (see [158]).

In section 6.2, the terms of *equilibrium*, *mispricing*, *arbitrage* together with those of *mean reversion*, *error correction* and *statistical arbitrage*, will be discussed in further detail. *Time-warped co-integration* is presented in parallel with the known co-integration theory and is evaluated through the *statistical arbitrage* framework as proposed by BURGESS et al. (1996), [24]. The discussion and outlook section, concludes this chapter with ideas for possible generalization extensions of the proposed methods.

6.1 Multivariate Pattern Matching

The question that initially arises when working with multiple data channels is which data sources should be employed for solving the problem and how many of those should be selected. This is surely a very difficult task considering the huge amount of data waiting for processing. Objective functions, therefore, become a necessity in addressing this problem. In relation to the above questions, objective functions may concern computational efficiency, avoidance of information repetition and degree of relevance of the data to the problem. Sophisticated computational and statistical techniques aim to optimise these objective functions. Correlation analysis [19], Reduced Autocorrelation Modelling [87], Principal Component Analysis (PCA) [83], Independent Component Analysis (ICA) [80, 81], Singular Value Decomposition (SVD) [23, 141] and co-integration [52] are modern statistical techniques applied in controlling the dimensionality of multivariate problems under the constraint of keeping substantial amounts of information. Utans et al., (1997), [175], showed that it is possible to reduce the number of input series to a model of hourly exchange rates using principal component analysis to reveal the main uncorrelated factors driving the market. However, extracting uncorrelated time series which are also independent and identically distributed (IID), is not always possible. HSIEH, (1988), [76], identified for example, that although price changes of a set of currencies are uncorrelated, they are not IID. One year later, [77], he demonstrated non-linear dependence in daily exchange rates using non-linear stochastic functions. Here, a rather simple, two stage method is used for data set selection and dimensionality reduction.

6.1.1 Data Selection

To achieve compatibility with the rest of the thesis, the data set to be processed consists of the series used in the previous chapters. Thus, we investigate the forecasting and trading ability of the pattern matching method on the multivariate set of market indices of the following European countries: UK, France, Germany, Spain, Italy, Greece and Switzerland. The selection of the input data is performed in two stages. Firstly, a set of indices that meet specific constraints, eg. stock indices that are strongly affected by a common economic policy, European indices, are *a-priori* selected. Secondly, the selected data set is processed using Principal Component Analysis in order to reveal data sources that do not sufficiently contribute to the overall variance of the underlying system. In effect, dimensionality reduction of the input is achieved and also correlation information on the data set is statistically justified.

Principal Component Analysis (PCA) Principal Component Analysis (PCA) is an advanced technique for extracting structure from possibly high-dimensional data sets. It is readily performed by solving the *eigenvalue* problem, or by using iterative algorithms which estimate *principal components*. For reviews of the existing literature, see JOLLIFFE, (1986), [86] and DIAMANTARAS et al., (1996), [49]; some of the classical papers on PCA are due to PEARSON, (1901), [130]; HOTTELLING, (1933), [75]; KARHUNEN, (1946), [89]. PCA is an orthogonal transformation of the coordinate system in which we describe the data. The new coordinate values which represent the data are called *principal components*. It is often the case that a small set of principal components is sufficient to account for most of the structure of the data. These are sometimes called factors or latent variables of the data.

The present work studies PCA in the case where we are not interested in principal components in input space, but rather in correlations of the principal components with the input space. In the next, we will first review the standard PCA algorithm and give the formulas for calculating the correlation coefficient between the ith input variable and the jth principal component.

Given a set of centered[2] input vectors $\mathbf{x}^n = \{x_1^n, x_2^n, \cdots, x_i^n, \cdots, x_d^n\}$, $i = 1, \cdots, d$, $n = 1, \cdots, N$, PCA aims at a linear transformation of the original input variables $\mathbf{x}^n$ that results d uncorrelated - orthogonal - variables $\mathbf{z}^n$, the *principal components*. The transformation matrix S (covariance matrix) of the set of vectors $\{\mathbf{x}^n\}$ is:

$$S = \sum_{n=1}^{N} (\mathbf{x}^n)(\mathbf{x}^n)^T \quad . \tag{6.1}$$

PCA diagonalizes the covariance matrix S by solving the eigenvalue equation:

$$S\,\mathbf{u}_i = \lambda_i\,\mathbf{u}_i \quad , \tag{6.2}$$

where $\mathbf{u}_i$, $i = 1, \cdots, d$ are the eigenvectors and λ_i the eigenvalues of the covariance matrix S[3]. The transformed variables are refered to as the principal components - *t-scores* - and are ordered by explained variance:

$$var(z_1^n) \geq var(z_2^n) \geq \cdots \geq var(z_d^n) \quad . \tag{6.3}$$

The variances of the principal components are given by the eigenvalues of the covariance matrix. The eigenvectors of the covariance matrix are refered to as the *loadings* and they indicate the contribution of the original variables to the system's variance.

In this work, we are interested in the correlation coefficients between the input and the principal component variables. It is possible to determine the correlation of the ith principal component and the jth original variable using the following formula:

$$r_{ij} = \frac{u_{ij}\sqrt{\lambda_i}}{s_j} \quad , \tag{6.4}$$

where u_{ij} are the loadings (eigenvectors of the covariance matrix), λ_i the eigenvalues - the contribution in terms of variance of the principal components to the overall system and s_j is the variance of the original input variable $\mathbf{x}_j^n$ (Here, $s_j = 1$ because data has been normalised to unit std). The original time series variables which are highly correlated to the first principal component are the ones which construct the multivariate data set on which the pattern matching and prediction algorithm is applied.

[2]The time series signal is normalized to zero mean and unit standard deviation

[3]A comprehensive proof of the PCA algorithm can be found in [17]

6.1.2 Matching Forecasting

Having the multivariate data set selected using principal component analysis, the pattern matching and prediction method, presented in the previous chapter, is employed in its generalized form for the multivariate case. On segmented time series the 'current' pattern is selected from the series to be forecasted using the segment embedding dimension technique of section 5.3.1. This pattern is matched on the whole multivariate set at several historical positions so that a vector of historical matched patterns is extracted. The averaged movements that follow those historical matched patterns, after normalization in terms of level and duration, give the predictive indication for the series to be predicted. Because this is a simple generalization of the pattern matching and prediction univariate approach, we refer the reader to the previous chapter. What we want to emphasise here is that the employment of additional time series for the forecasting task, may improve the performance of the pattern matching and prediction algorithm. Before presenting the forecasting and trading results of the simulations done using this approach, we must state that the improvement achieved does not suffer from any substantial increase in the computational time. The computational time increases linearly as more than one series is added to the prediction system.

PCA is applied to the selected data set to calculate the correlation coefficient of equation 6.4. The market index time series that are highly correlated to the first principal component are selected as the winners and are the ones to which the pattern matching and prediction algorithm is applied. As seen in table 6.1, the German, French, British and the Spanish market indices are those highly correlated to the first principal component.

Each time we predict one market index using all the others from the data set. For each simulation 60% of the data is used as the historical part of the time series needed for pattern matching. Therefore, the forecasting performance for a segment ahead is tested together with the trading performance of the system on the remaining 40% of the data. This corresponds to almost 1100 points of daily prices, or to a four years period. The trading strategy is the same as the one used in evaluating the univariate case (see section

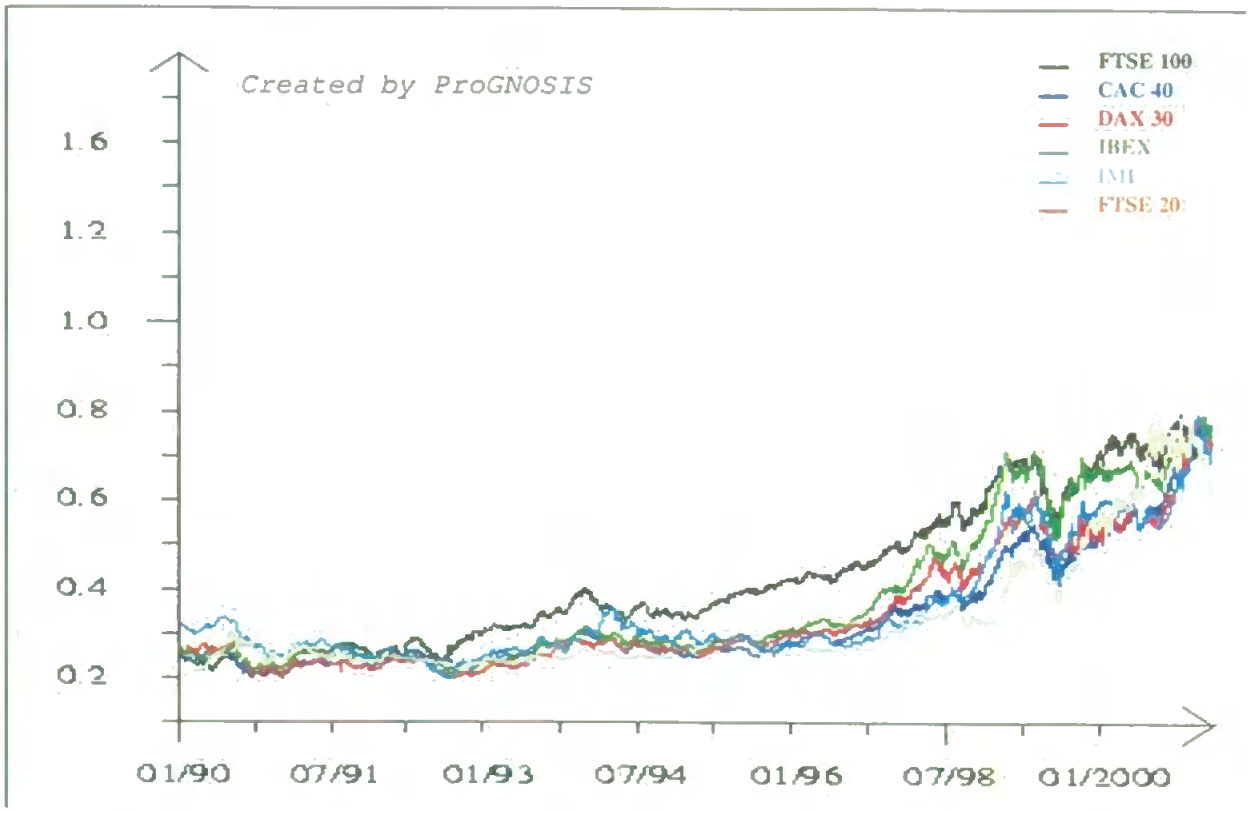


Figure 6.1: The European market indices (01.01.1990 - 26.04.2000) that were used for experiments in this thesis. All series are scaled to lie between 0.2 and 0.8.

5.3.3) That allows straight forward comparison to the results presented in section 5.4.

In table 6.2, it can be seen that there is not a significant improvement in the forecasting direction ability of the multi-pattern matching system compared to the univariate one. In addition similar performance is recorded for the case when market indices which are characterised by high correlation coefficients are employed for the multivariate prediction task. The average trading performance of the system remains positive for the multivariate pattern matching system. The auto-trade component of PROGNOSIS on multivariate data, however did not significantly outperform the univariate one on the market indices used in this study. Note that brokerage costs of 1% are also considered in our calculations. Overall, a substantial improvement in the performance of the algorithm was not achieved after applying PCA. However, computational time was saved.

	UK	FRANCE	GERMANY	SPAIN	ITALY	GREECE
r	0.83	0.87	0.94	0.78	0.73	0.69

Table 6.1: The correlation coefficient, r of each of the european DATASTREAM market indices to the corresponding 1st principal component as calculated from equation 6.4. The original time series are daily prices between January 1990 and April 2000. A predefined threshold of 75% is set for selecting time series from the data set for dimensionality reduction.

market index	*pred (%)*	*profit*	*SR*	*pred* (%)*	*profit**	*SR**
UK-DS-MK	58.70	10.1	1.46	59.35	7.64	0.78
FRANCE-DS-MK	60.25	6.47	0.61	61.30	11.02	1.87
GERMANY-DS-MK	60.00	12.2	1.34	60.65	8.43	1.36
SPAIN-DS-MK	58.38	15.58	2.31	56.83	13.62	2.09
ITALY-DS-MK	54.86	3.36	0.70	59.68	8.30	1.36
GREECE-DS-MK	56.75	7.11	0.87	55.45	7.61	0.95

Table 6.2: Prediction and trading results using multivariate pattern matching. The series are between 01.1990 and 04.2000. All simulations have been performed for 60% of the data set and tested on the remaining 40%. In the columns indicated by $*$, the results obtained without using PCA are given (all the series are used). *SR* stands for Sharpe Ratio, a risk measurement which accompanies the trading performance. Profit is measured in basis points.

6.2 Time-warped co-integration

In this section, we will discuss a novel way of exploiting long term relationships using the theory of co-integration (ENGLE and GRANGER (1987) [52]; HARGREAVES, (1994), [72]; MADDALA et al., (1998), [109]) between economic and financial time series. Local time dependencies between the time series in this framework are taken into account to test their co-integrating relationships. Intuitively, the reasoning behind this approach is that non-stationarities in the residuals of two time series may occur due to the fact that the differences of the time series points are taken synchronously. This means that each point of the residual series is calculated by differencing the prices of the original series X, Y that are identified by the same time index ($r(t) = x(t) - y(t)$). Local time shifts or delays between the time series introduce data point projections between the time series that do not occur synchronously. A data point of the series X at time t may be projected to others of series Y that occur forwards or backwards in time, i.e. at time $t \pm \alpha$, $\alpha > 0$. In addition, a point of series X may be projected to more than one point of series Y. This corresponds to local time stretches or shrinks of one time series relative to the other. Such data point projections may identify local time dependencies between the time series. Furthermore, a projection between two time series X, Y that reveals local time dependencies may overcome the **spurious regression** problem[4] and can lead to a form of **co-integration**. The economic time series regression literature ([98, 42, 48]) suggests that the one time that the spurious regression problem can be surmounted is when two time series X, Y, are co-integrated. This concerns the errors of the following regression model:

$$e_t = Y_t - \alpha - \beta X_t \tag{6.5}$$

In particular, if the regression error series (also known as the residual series) that results from the regression equation 6.5 exhibits stationary behaviour (does not have a "unit root"), it is said that X, Y are co-integrated.

[4]If two time series X, Y are non-stationary, i.e. they have a "unit root", then all the usual regression results might be misleading and incorrect. This problem is known as the spurious regression problem.

However, it is expected that the residuals series of two non-stationary time series will also be non-stationary. Statistically, it is this non-stationarity that causes the spurious regression problem mentioned earlier. What we want to investigate here is whether any local time dependencies (shifts, delays, shrinks, or stretches) identified between two time series, can reveal stationary residual series. To test this, we apply the Dynamic Time Warping (DTW) algorithm, [99, 14] (see section 5.2), to the time series in order to identify local time dependencies and to extract the residual series based on those dependencies. In particular, DTW is used to discover an optimal projection between the time series. Details on this approach together with explanatory figures and examples are presented later in this chapter.

Stationarity in the residual series introduces a co-integrating relationship between the original time series. Because of the time relations that are identified between the time series, we call this type of co-integration, **time-warped co-integration**. This is discussed in section 6.2.3. Within this framework, the mean reversion effect, [6, 94, 140], of the residuals of co-integrated series is also discussed as a consequence of the proposed time-warped co-integration relationship (section 6.2.1). Basically, *mean reversion* constitutes a forecasting indication for the future movement of the equilibrium error series (residuals). It indicates that when two series are linked with co-integration, indeed, their equilibrium error series will sooner or later return near the zero line. This constructs a '*fair price*' relationship for the time series. To further support the forecasting indication of the *mean reversion* effect and to check when the "*fair price*" is likely to occur, a forecasting mechanism based on historical pattern matching predicts the next values of the residual series as well as the time movement of one series relative to the other. These predictions drive the timing of the trading system, which operates as a technical arbitrageur and takes opposite trading positions on the co-integrating financial time series. We evaluate the proposed system, in terms of trading performance, within the "*statistical arbitrage*" framework, as discussed by BURGESS, (1997), [25].

Before exploring these ideas in more depth, we give here some initial visual intuition about the time series local time dependencies and the time series points projection that

identifies them. In figure 6.2 the local time dependencies are demonstrated on simulated data. Local time delays and shifts have been inserted into two perfectly co-integrated series. The DTW based algorithm, when applied to these two time series, managed to almost perfectly recover all the locally inserted shifts and delays (see 2nd row of the figure).

In figure 6.3, the same procedure has been applied to real financial time series. In this figure the equity series of BARCLAYS and HSBC are drawn from January 1999 until January 2000 in daily closing prices. These time series were picked randomly for illustration purposes. The local time projection of the series and a new time series representation that derives from this projection are also depicted there. It is very difficult to draw any valid conclusions out of these figures about the local time dependencies between the series. However, our aim here is only to give some visual intuition about this. Detailed evaluation, both statistical and in terms of trading performance is given later in this chapter.

In the following, we first address the basic attributes of co-integration as an econometric theory and consequently the co-integration mean reversion effect (section 6.2.1). Some stationarity issues are addressed in section 6.2.2. Time-warped co-integration and the dynamic time warping mechanism that retrieves time dependencies between two series is discussed in section 6.2.3. In section 6.2.4, the algorithm that predicts the way that the time series will move relative to each other is presented. The model is systematically evaluated through profit measures, calculated with a statistical arbitrage trading strategy, in section 6.2.5. Finally, we summarise the main aspects of the proposed systems and we conclude with future research ideas (section 6.2.6).

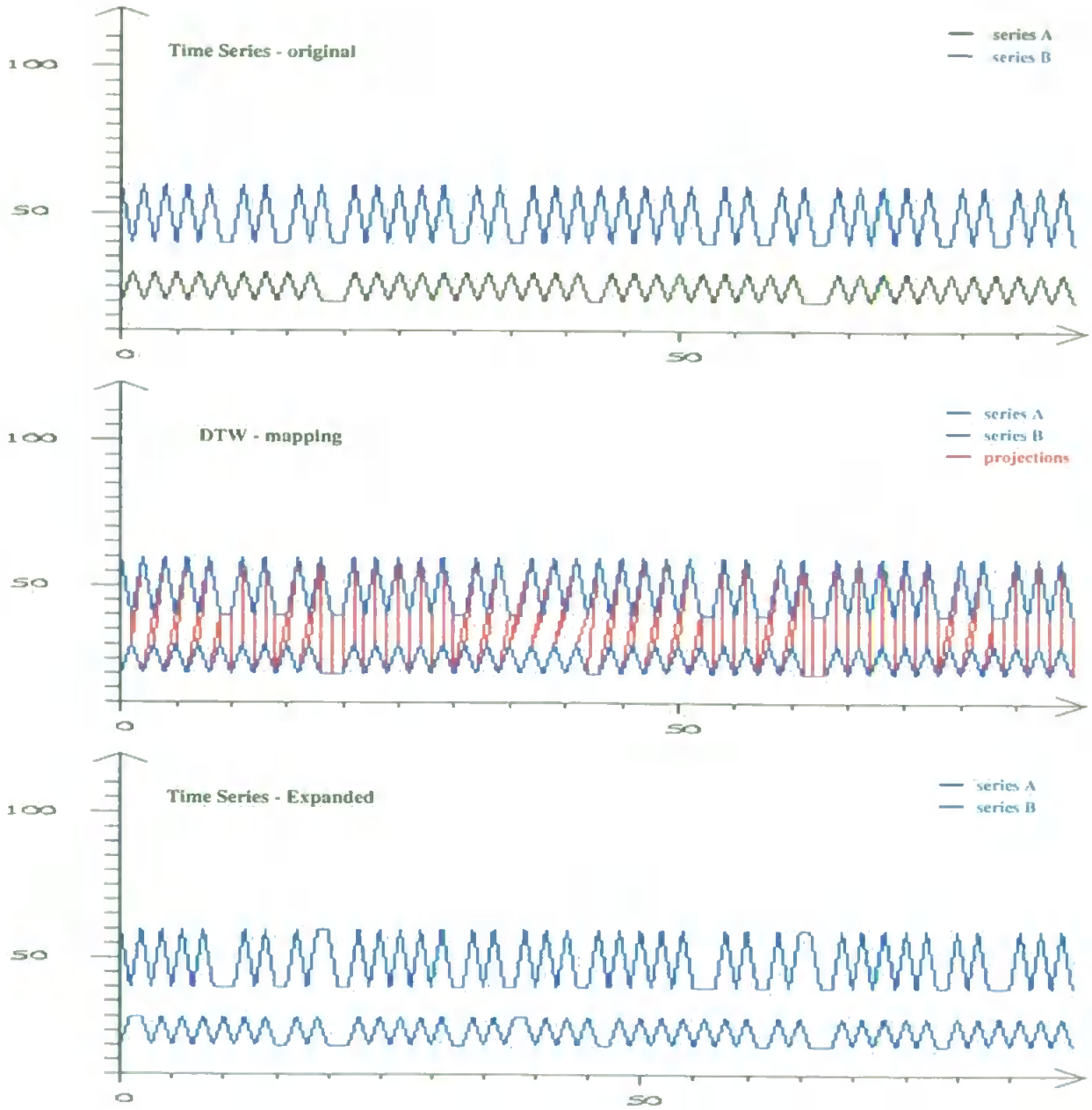


Figure 6.2: Introducing time delays or shifts in the relation between simulated data. **From top to bottom: Top:** Original time series (with time shifts and delays). **Middle:** The time series projection using dynamic time warping. **Bottom:** Expanded time series.

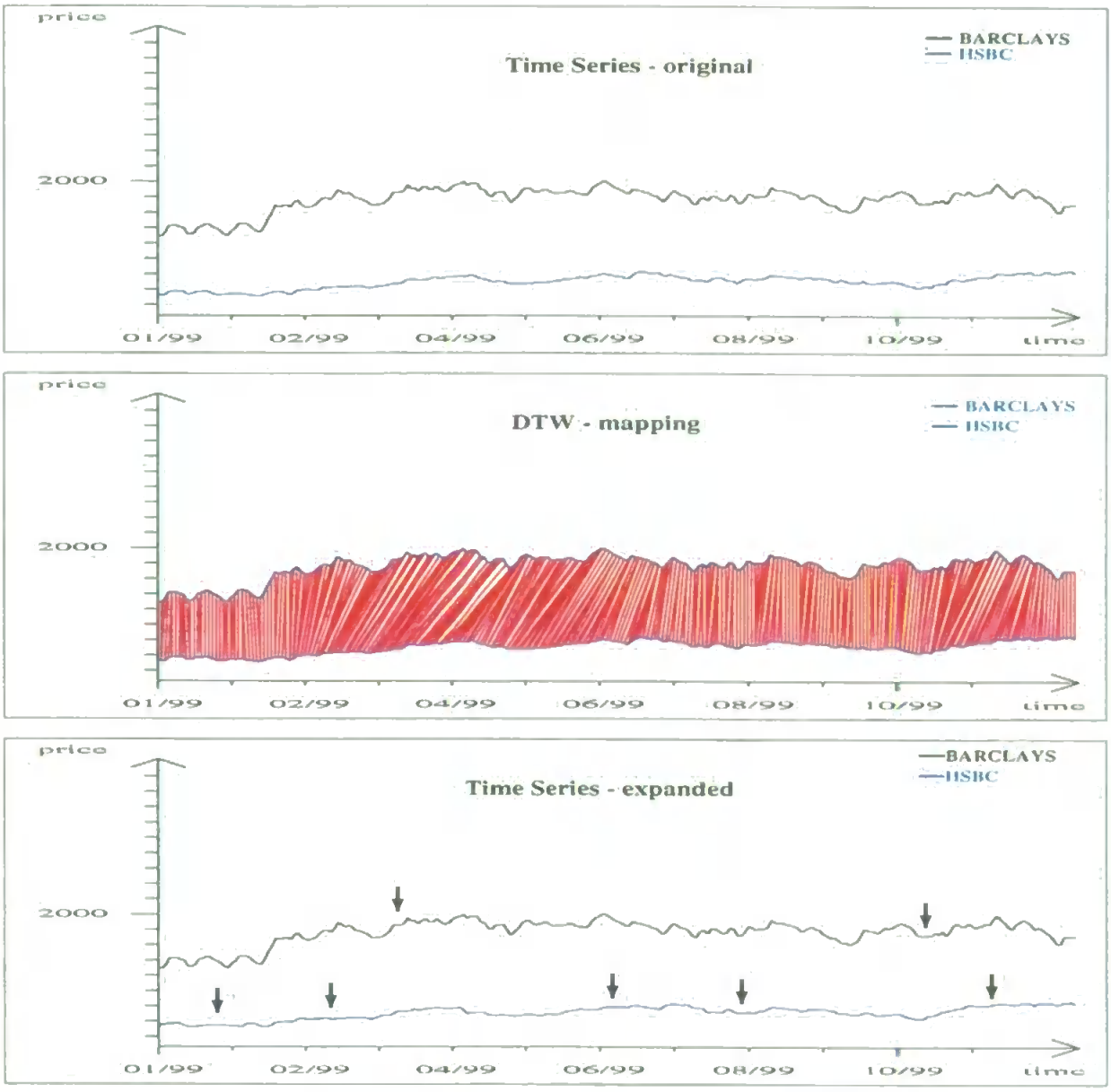


Figure 6.3: Introducing time delays or shifts in the relation between two stock from the UK bank sector. Here we depict the BARCLAYS and the HSBC stocks for 1999 (01.99-12.99) **From top to bottom:** 1. Original time series. 2. DTW mapping between the series 3. Time series after local expansion (The arrows on the figure point to local expansion positions).

6.2.1 Co-integration

The econometric concept of co-integration Co-integration initially arose from the concept of **Error Correction Models (ECM)**, which dated back to the paper by SARGAN, (1964), [154]. Sargan in this paper, studies non-stationarities and spurious correlation between wages and price time series in the UK. Generally, co-integration is based on the idea that two financial time series are moving relative to each other. That is often refered to as the **equilibrium relationship** and in its simplest form is given by the following equation:

$$y_t = \alpha + \beta\, x_t \quad . \tag{6.6}$$

$$v_t = y_t - \alpha - \beta\, x_t \quad , \tag{6.7}$$

is the mathematical formula which writes the **disequilibrium error** as a linear combination of x_t and y_t.

ENGLE and GRANGER in 1987, systematically defined the concept of co-integration by the following:

"If a long-run equilibrium relationship such as in equation 6.6 exists, then disequilibrium errors should form a stationary time series and have zero mean".

In econometric terms, that is v_t in equation 6.7 should be integrated of order zero[5], and $E(v_t) = 0$.

To summarise this, we can say that the time series x_t and y_t are co-integrated, if both the time series become stationary on first differencing, i.e. are integrated of order one and their disequilibrium error linear relationship (equation 6.7) is stationary, i.e. integrated

[5]Consider two time series x_t, y_t which are integrated of order d, i.e. become stationary after differencing d times. If the parameter set (α, β in equation 6.6) of the linear transformation of x_t, y_t is such that the long-run changes in level of x_t, y_t cancel out, i.e. are approximately the same, then it is possible that $v_t = y_t - \alpha + \beta\, x_t$ is integrated of order $d - b$, for $b \geq 1$. In our example, we take x_t, y_t to be integrated of order one and we expect the disequilibrium error series to be integrated of order zero, i.e. stationary in level, for the long-run co-integration relationship between x_t and y_t to exist.

of order zero, for a parameter set α, β. This theorem is of great value in econometrics because it has often been stressed that economic time series become stationary after first differencing. In other words, economic time series often may be *mean reverting*.

Mean Reversion "*Mean reversion refers to a tendency of asset prices to return to a trend path*" (BALVERS et al., (2000), [6]). By itself this definition points out the forecasting character on the mean reversion effect. It has been said earlier that mean reversion indicates a '*fair price*' relationship between two co-integrated financial time series. This relationship derives from the tendency of the residuals (disequillibrium) series to return near the zero line. Empirical evidence for the presence of mean reversion over long horizons in the US stock market has been provided first by FAMA et al., (1988), [55] and POTERBA et al., (1988), [140]. Others like LO and MACKINLAY, (1988), [104] and KIM et al., (1991), [94] argue for the absence of the mean reversion effect on US stock prices using weekly data and conclude that mean reversion may occasionally occur in selected time periods but not broadly. Between those contradictory opinions, we quote CAMBELL et al., (1997), [28] saying:

"*Overall, there is little evidence for mean reversion in long horizon returns, though this may be more of a symptom of small sample sizes rather than conclusive evidence against mean reversion - we simply cannot tell.*"

Therefore, we have taken mean reversion on financial markets as an open research area upon which to base our work in this section. Here, it is assumed that the residual series is mean reverting, if it does not have a "unit root"[6], i.e. it is stationary. Due to local time dependencies that are embedded in the residual series calculation, we call this type of mean reversion, **time-warped mean reversion**.

As it will be shown later in this section, local time dependencies of economic time series reveal considerable correction in terms of co-integrating movement. To test this hypoth-

[6] "Unit root" tests are tests for time series stationarity. The most well known "unit root" tests are the Dickey-Fuller (DF), Augmented Dickey-Fuller (ADF), suggested by ENGLE and GRANGER (1987), [52] or the Durbin-Watson test proposed by SARGAN and BHARGAVA (1983), [155].

esis, we evaluate time-warped mean reversion through the statistical arbitrage trading strategy as defined by BURGESS, (1997), [25]. In the next (section 6.2.3), we define the time-warped co-integration, i.e. co-integration using local time shifts and delays.

6.2.2 Stationarity issues

A time series is said to be *stationary* if its mean, variance and covariances remain constant over time. Formally, *non-stationary* merely means *anything that is not stationary*. However, a time series is non-stationary if it fails to satisfy any part of the definition above.

Financial analysts usually focus on a particular type of non-stationarity that seems to be present in many financial and macroeconomic time series: "**unit root**" non-stationarity (see [72, 170]). The "unit root" stationarity rule says that if a time series has a "unit root" then it is non-stationary. Stationarity is exhibited in the opposite condition.

DICKEY and FULLER 1979, [51], first proposed a "unit root" test that overcomes the problems that arise when using autoregressive (AR) processes with OLS estimators to check for stationarity (see THOMAS 1997, [170]). Other known "unit root" tests for stationarity based on the residuals of the OLS regression are: the *Augmented Dickey-Fuller* (SAID and DICKEY 1984, [150]); the *Co-integrating Regression Durbin-Watson* (ENGLE and GRANGER 1987, [52], SARGAN and BHARGAVA (1983), [155]); the *Phillips* and *Phillips-Perron* (PHILLIPS 1987, [136] and PHILLIPS and PERRON 1988, [138]) etc.[7]

Here, we test for "unit root" stationarity of residuals series using the *t-statistic* (t-stat) values of the Dickey-Fuller test, [51]. Autoregressive (AR) models of the residuals series are estimated.

- If the final models include deterministic trend the Dickey-Fuller critical value for the t-stat is approximately -3.45 taking the level of significance of the regression to be 0.05

[7]Other stationarity tests based on principal components or canonical correlation have been developed by JOHANSEN (1988), [84] and PHILLIPS and OULIARIS (1988), [137].

- If the final models do not include deterministic trend the Dickey-Fuller critical value for the t-stat is approximately -1.65 taking the level of significance of the regression to be 0.05

If the t-stat on the first explanatory variable of the regression model is more negative than the proposed critical values, the "unit root" hypothesis is rejected and it is concluded that the residual series is stationary, [98].

In table 6.3, the t-stat values of the regression with no deterministic trend of the residual series that derive from six European DATASTREAM market indices (01.01.1998-26.04.2000) are presented.

	t-stat		t-stat
UK - FRANCE	-1.96	FRANCE - GREECE	-0.06
UK - GERMANY	-1.86	GERMANY - SPAIN	-2.39
UK - SPAIN	-2.19	GERMANY - ITALY	-2.92
UK - ITALY	-1.92	GERMANY - GREECE	-0.64
UK - GREECE	-2.71	SPAIN - ITALY	-3.44
FRANCE - GERMANY	-2.37	SPAIN - GREECE	-2.17
FRANCE - SPAIN	-1.82	ITALY - GREECE	-1.46
FRANCE - ITALY	-1.77		

Table 6.3: The t-stat values of the regression with no deterministic trend of the residuals series that derive from six European DATASTREAM market indices (01.01.1998-26.04.2000). The regression level of significance was 0.05. The critical t-stat value for the Dickey Fuller test is -1.65.

The next table, (table 6.4), depicts the t-stat values for the same data set after applying the DTW time series projection algorithm.

A comparison between the two tables shows that the DTW projected time series give more negative or at least similar t-stat values than the critical Dickey-Fuller ones for more than 60% of the time. In figure 6.4 a plot of the t-stat values given in tables 6.3 and 6.4

	t-stat		t-stat
UK - FRANCE	-2.42	FRANCE - GREECE	0.85
UK - GERMANY	-2.58	GERMANY - SPAIN	-2.96
UK - SPAIN	-2.57	GERMANY - ITALY	-2.56
UK - ITALY	-1.90	GERMANY - GREECE	-1.11
UK - GREECE	-2.22	SPAIN - ITALY	-4.08
FRANCE - GERMANY	-1.97	SPAIN - GREECE	-2.44
FRANCE - SPAIN	-0.59	ITALY - GREECE	-1.53
FRANCE - ITALY	0.13		

Table 6.4: The t-stat values of the regression with no deterministic trend of the residuals series that derive from six European DATASTREAM market indices (01.01.1998-26.04.2000) after they have been projected using DTW. The regression level of significance was 0.05. The critical t-stat value for the Dickey Fuller test is -1.65.

compared to the Dickey-Fuller critical value, is depicted. According to this the residuals of the DTW projected time series remain stationary if the original residual series are stationary too. This allows us to assume that co-integration theory can be applied to the

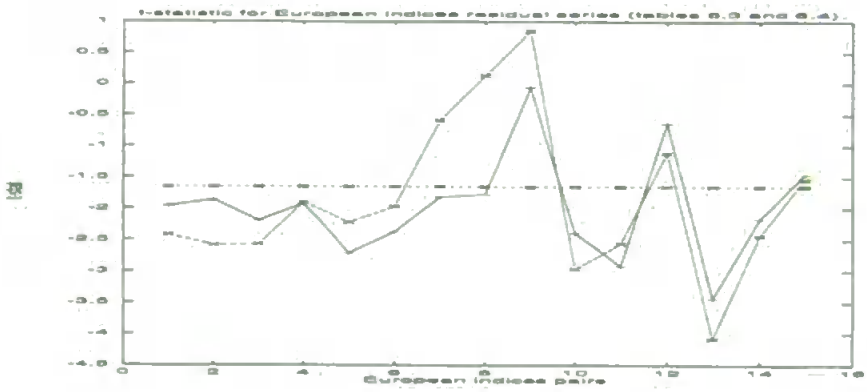


Figure 6.4: The t-stat values of the residual series of the European DATASTREAM market indices. This plot corresponds to tables 6.3 (solid line), 6.4 (dashed line). The straight line corresponds to the Dickey-Fuller critical value.

time series representations that include local time dependencies. In the next section, we explain how co-integration is considered in this context.

6.2.3 Dynamic Time Warping and Co-integration

In line with the established concept of co-integration, we make the following hypothesis:

"*Economic time series follow an equilibrium relationship which can be recovered when relative successive local time delays and shifts or shrinks and stretches of one series relative to the other are presumed*".

Under this hypothesis, a new form of equilibrium relationship and error correction model, using time delays and shifts information may be introduced.
In the simple case of two time series x_t, y_t which are taken to be integrated of order one, we write the following equilibrium state formula:

$$y_t = \alpha + \beta \frac{x_t + x_{t-1} + \cdots + x_{t-k_t}}{k_t + 1} \tag{6.8}$$

where k_t is a time varying parameter which introduces higher order lags in the equilibrium relationship. Note that for $k_t = 0$, the equilibrium relationship of equation 6.6 can be reproduced. Following the terminology of the previous section, we can also write:

$$v_t = y_t - \alpha - \beta \frac{x_t + x_{t-1} + \cdots + x_{t-k_t}}{k_t + 1} \tag{6.9}$$

Equation 6.9 stands for the disequilibrium error series, which is the linear combination of y_t and a vector of x past values, $\{x_t, x_{t-1}, \cdots, x_{t-k_t}\}$.
The key idea behind this new equilibrium relationship is the estimation of the parameter k_t, which varies over time. k_t is the parameter that indicates how far to look in the past of a series X in order to explain its relation to the current situation of series Y and vice versa. A graphical explanation of this idea is given in figure 6.5.

In that figure we show that if a mapping that extracts the connections between relevant points of the series X, Y can be identified in such a way that represents the relative successive time delays or shifts between them, then a more realistic error correction model

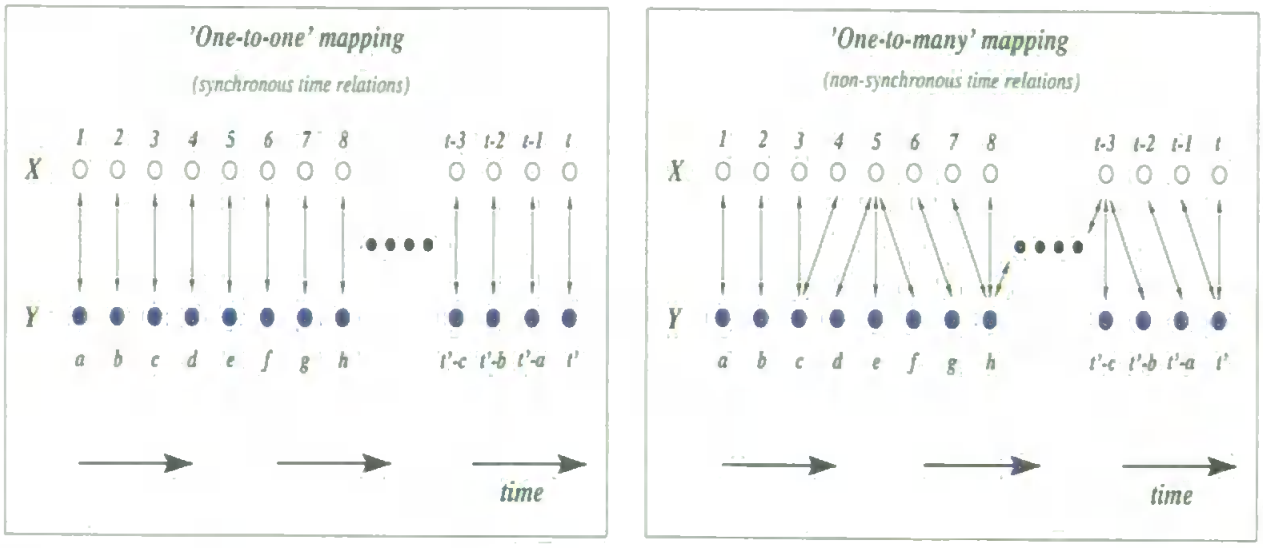


Figure 6.5: Mapping between two time series, X, Y. **Left:** The usual "*one-to-one*" time series mapping with no time relations. **Right:** The proposed "*one-to-many*" mapping with time relations. An illustration of relative time delays and shifts.

can be designed.

A method to achieve that kind of mapping is the Dynamic Time Warping (DTW) algorithm. As described in section 5.2, DTW can be used for comparing time series because it optimally aligns the time series. This is realized through the warping path of the DTW-matrix, which is optimally calculated via dynamic programming. In figure 6.6, the warping path of the example series X, Y is drawn. There, it can be seen that the warping path can clearly identify the positions where one time series is locally delaying or shifting against the other (moving across a column or a row in the DTW-matrix).

In figure 6.6, it can been seen that an individual point of series X, eg. x_5, may be projected on more than one point of series Y, (y_d, y_e, y_f) and vice versa, i.e. x_5 is not only related in terms of co-integation, to its time synchronous partner point y_e but also to other neighbouring points of series Y, (y_d, y_f). Such projections on the DTW-matrix reveal local delays and shifts between two time series X, Y.

Before demonstrating realistic examples, using DTW type time series projections, we must define k_t of equation 6.8 in a systematic way.

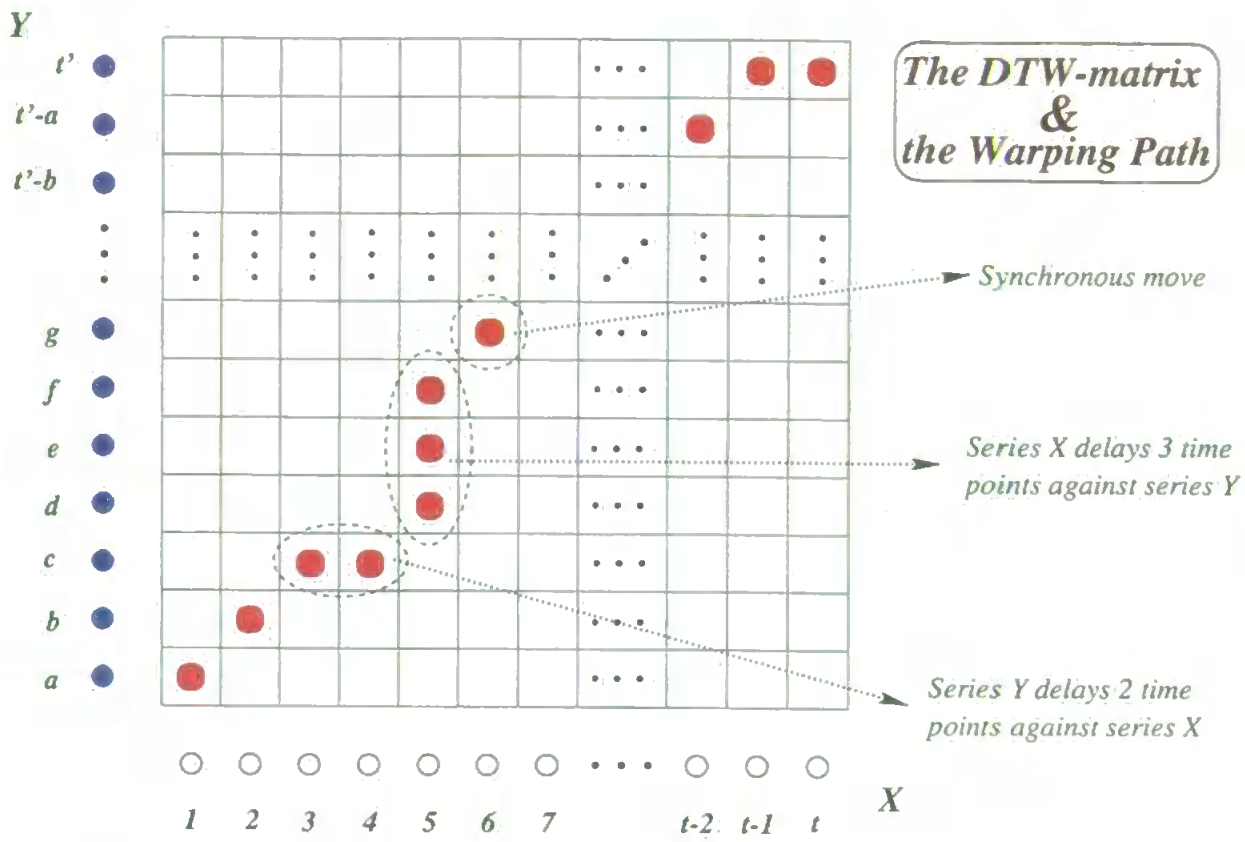


Figure 6.6: The optimum warping path, calculated via dynamic programming, is used to identify the local time delays and shifts between the series X, Y. Moves along columns of the DTW-matrix reveal delays of series X against Y, while those along rows reveal delays of series Y against X. Moving across the diagonal identify synchonous time series movement.

DEFINITION: *k_t at time t, when measured regarding one of the series in process, is the number of successive points that are placed in the column of the DTW-matrix indexed by the same time index t* (see figure 6.6).

By applying the above definition on the example of figure 6.6, we can write the values of k_t shown in table 6.5, together with the corresponding forms of their equilibrium function.

time t (ref. X)	k_t *values*	*Equilibrium Relationship*
1	0	$x_1 = \alpha + \beta\,[y_a/(0+1)]$
2	0	$x_2 = \alpha + \beta\,[y_b/(0+1)]$
3	0	$x_3 = \alpha + \beta\,[y_c/(0+1)]$
4	0	$x_4 = \alpha + \beta\,[y_c/(0+1)]$
5	2	$x_5 = \alpha + \beta\,[(y_d + y_e + y_f)/(2+1)]$
6	0	$x_6 = \alpha + \beta\,[y_g/(0+1)]$
.	.	.
.	.	.
.	.	.
t-3	1	$x_{t-3} = \alpha + \beta\,[(y_{t'-c} + y_{t'-b})/(1+1)]$
t-2	0	$x_{t-2} = \alpha + \beta\,[y_{t'-a}/(0+1)]$
t-1	0	$x_{t-1} = \alpha + \beta\,[y_{t'}/(0+1)]$
t	0	$x_t = \alpha + \beta\,[y_{t'}/(0+1)]$

Table 6.5: k_t **values and the equilibrium relationship** - k_t varies over time and the so does the equilibrium relationship. The values for k in this table correspond to the warping path depicted in figure 6.6.

Finally, to calculate the difference series after projecting the time series X, Y using dynamic time warping, we need to locally expand or compress (shrink or stress) them. This is shown in figure 6.7. In this thesis the expanded time series version is solely used. Extracting expanded representations of the time-warped co-integrated time series allows a "*one-to-one*" mapping between the series, which is compatible to the classic co-integrating time series theory. This, therefore, allows the usage of conventional co-integration evaluation technology, like arbitrage trading, on time-warped co-integration statistics and helps the user to monitor their performance in parallel[8].

[8]It is not very clear that the stationarity of the two time series is preserved under the DTW transformation. This point presumes future research

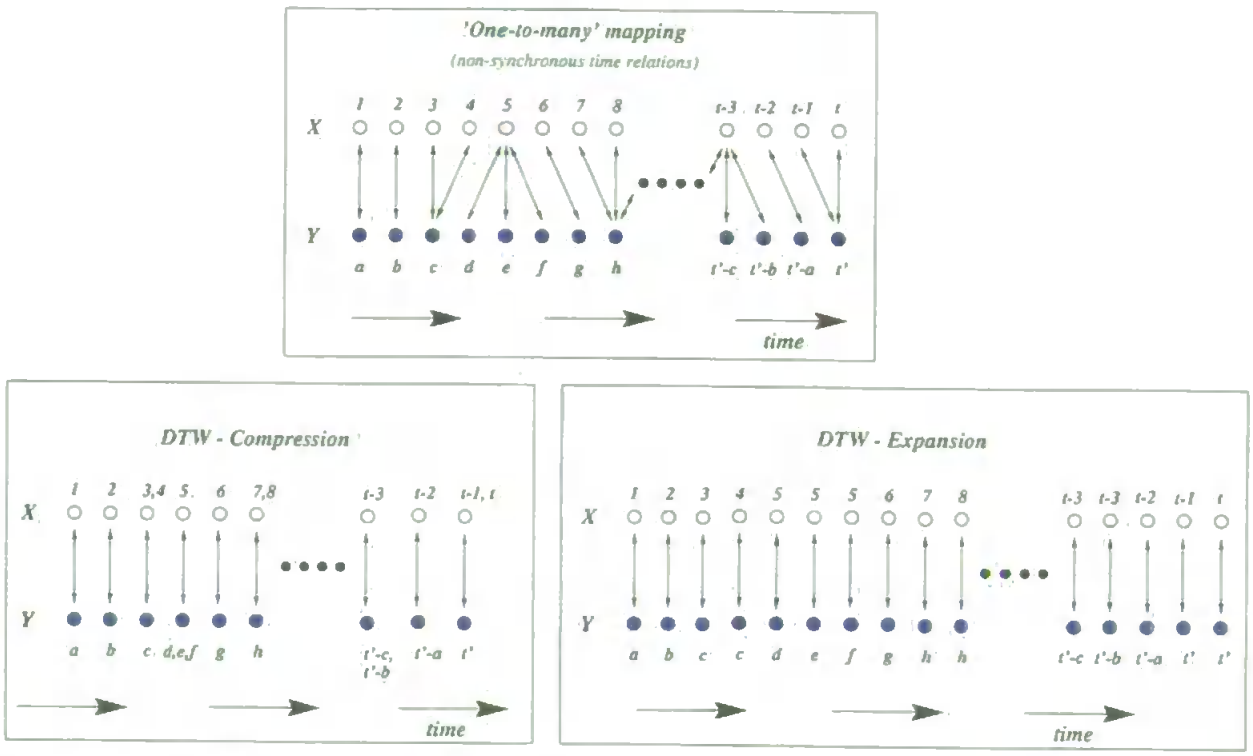


Figure 6.7: According to figure 6.5, we show here how the "one-to-many" mapping can be seen as 'one-to-one' mapping after locally compressing or expanding the time series. Note the decrease or increase of the number of points of the compressed or expanded time series.

We apply the Dynamic Time Warping based co-integration approach on a set of European stock indices. For clarification purposes, short duration parts of three European indices are depicted in figures 6.8, 6.9, 6.10. In these figures, the original series are depicted together with the projection lines which represent the time dependencies that are retrieved from dynamic time warping. Additionally, to each block of figures, the DTW path and the expanded version of the series are shown too.

We expect that through our *time-warped co-integration* time series study, '*equilibrium will occasionally occur at least to a very close approximation*', [52]. In effect, "mispricing" conditions may be detected and can be used for trading opportunities for "*statistical arbitrage*" (BURGESS and REFENES, (1996), [24]). Our effort is to evaluate the model through the statistical arbitrage trading strategy and directly compare our results with already published ones. Before that though, another important component of the pro-

posed system has to be discussed. This is the forecasting component which predicts the disequilibrium error value itself as well as local relative delays of one series against the other. This is achieved by predicting the warping path value as well as its movement within the DTW-matrix.

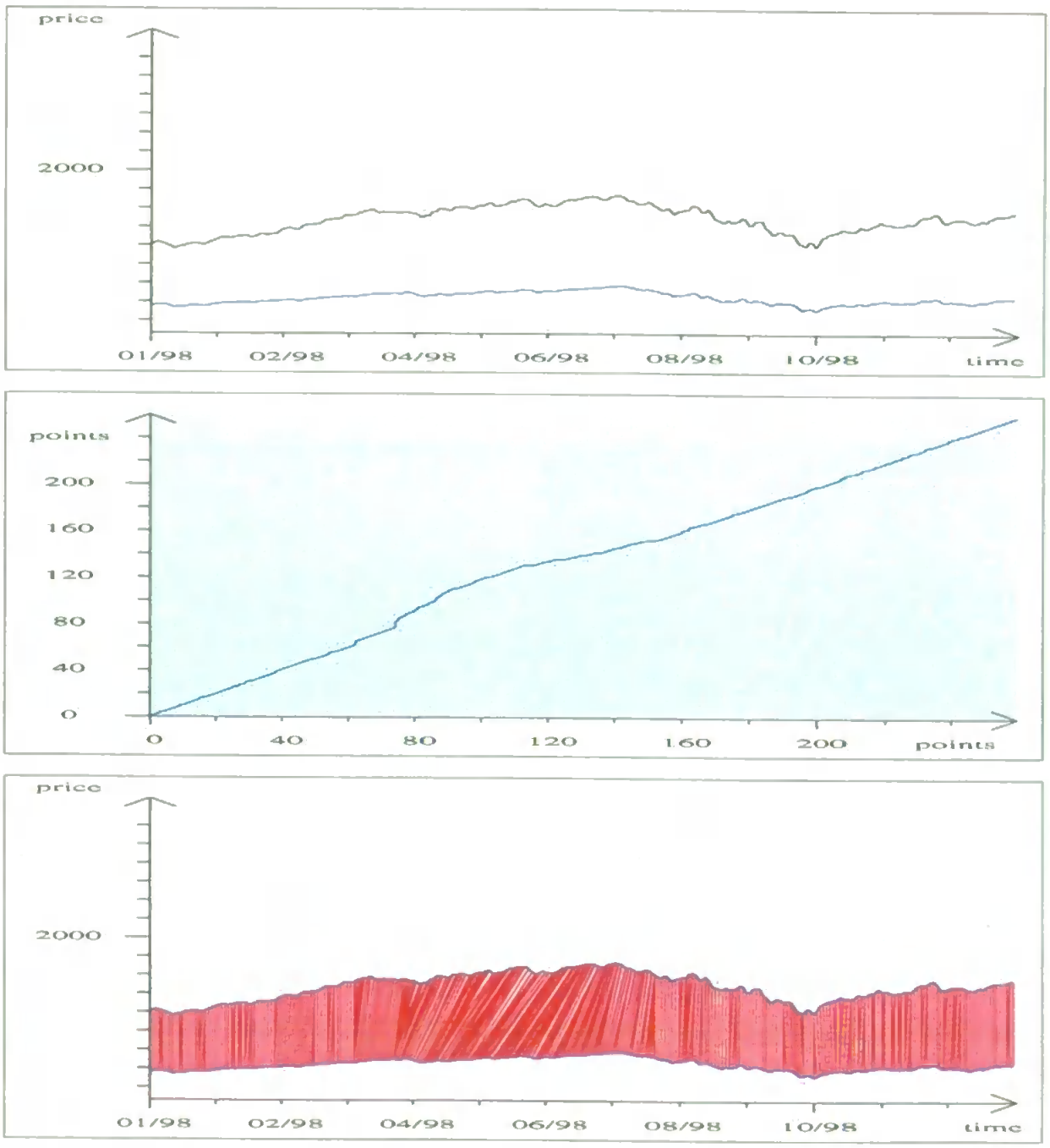


Figure 6.8: France - Germany DATASTREAM market indices. Profit: Arbitrage = -14.09, Arbitrage with DTW = 33.85

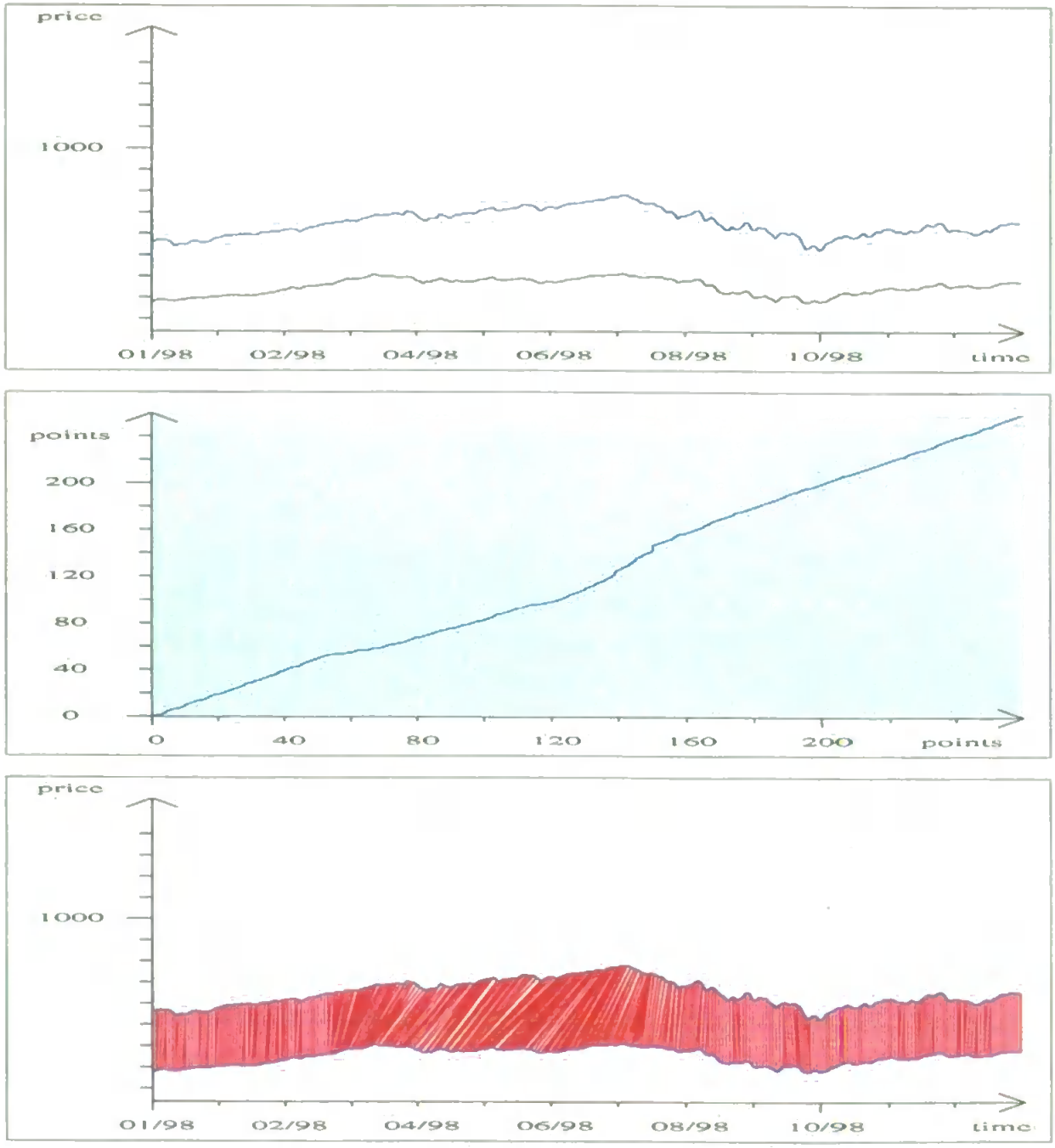


Figure 6.9: Germany - Spain DATASTREAM market indices. Profit: Arbitrage = 0.26, Arbitrage with DTW = 54.18.

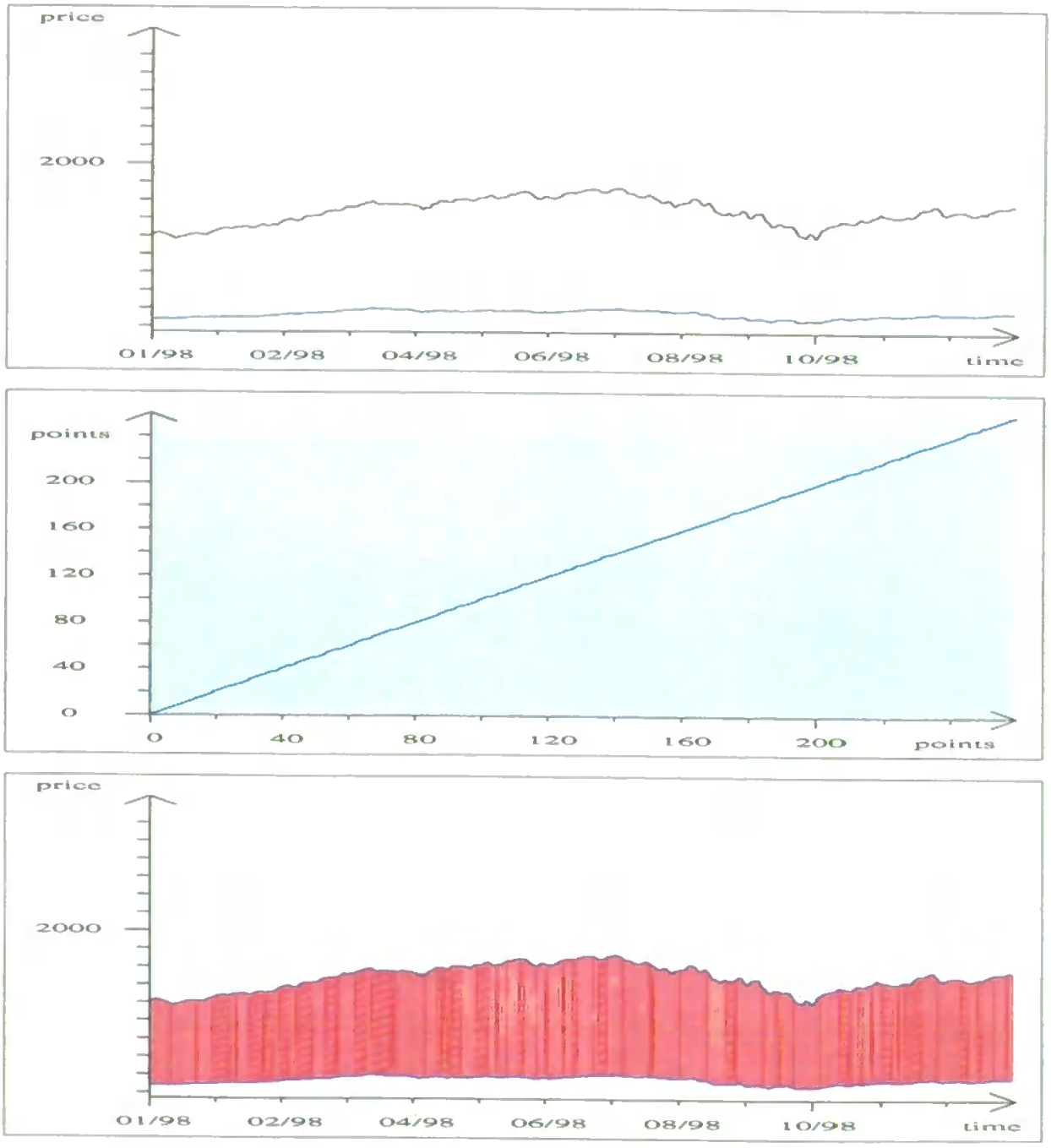


Figure 6.10: France - Spain DATASTREAM market indices. Profit: Arbitrage = -15.84%, Arbitrage with DTW = -16.38%

6.2.4 Predicting the Warping Path

To integrate forecasting power into the dynamic time warping co-integrating scheme, we present here a prediction model whose aim is to capture part of the deterministic behaviour in the mispricing dynamics. Two major prediction components are contained within this prediction model. The first provides forecasting indications about the changes of the actual prices of the linear combinations of the time series, that is the prices of the disequilibrium error series, which define the co-integration relationship. The second provides forecasting indications about the relative local time movements - local delays or shifts - of the time series. The latter forecasting component is vital for the estimation of the future equilibrium "*fair price*" relationship to be calculated.

This multicomponent prediction task may be achieved, when predicting the values of the cumulative distances corresponding to the warping path and its movement within the DTW-matrix. Here, we call the cumulative distances the magnitude of the warping path. The prediction model employed, is the one proposed in chapters 4 and 5. This model, adjusted for the warping path prediction task, is based on the idea of predicting the future movement and magnitude of the warping path using past warping path patterns which are similar to its 'current' situation. Figure 6.11 conveys that schematically.

To each node of the warping path three values are attached, its coordinates in the DTW-matrix and the value that corresponds to the cumulative distance measure between the ith and j^{th} price of the series X, Y respectively. The prediction output of the pattern matching prediction algorithm, therefore, is of three values. Apart from the residual disequilibrium error that is predicted, the next step movement of the warping path in the DTW-matrix is also given. This indicates whether series X will move forward relative to series Y or vice versa. Remember here, that delays of one series against the other in the DTW-matrix are represented with movements of the warping path along one column or row of the matrix. Diagonal steps in the matrix correspond to synchronous movements (see figure 6.6).

Before applying the pattern matching prediction algorithm, a transformation of the two

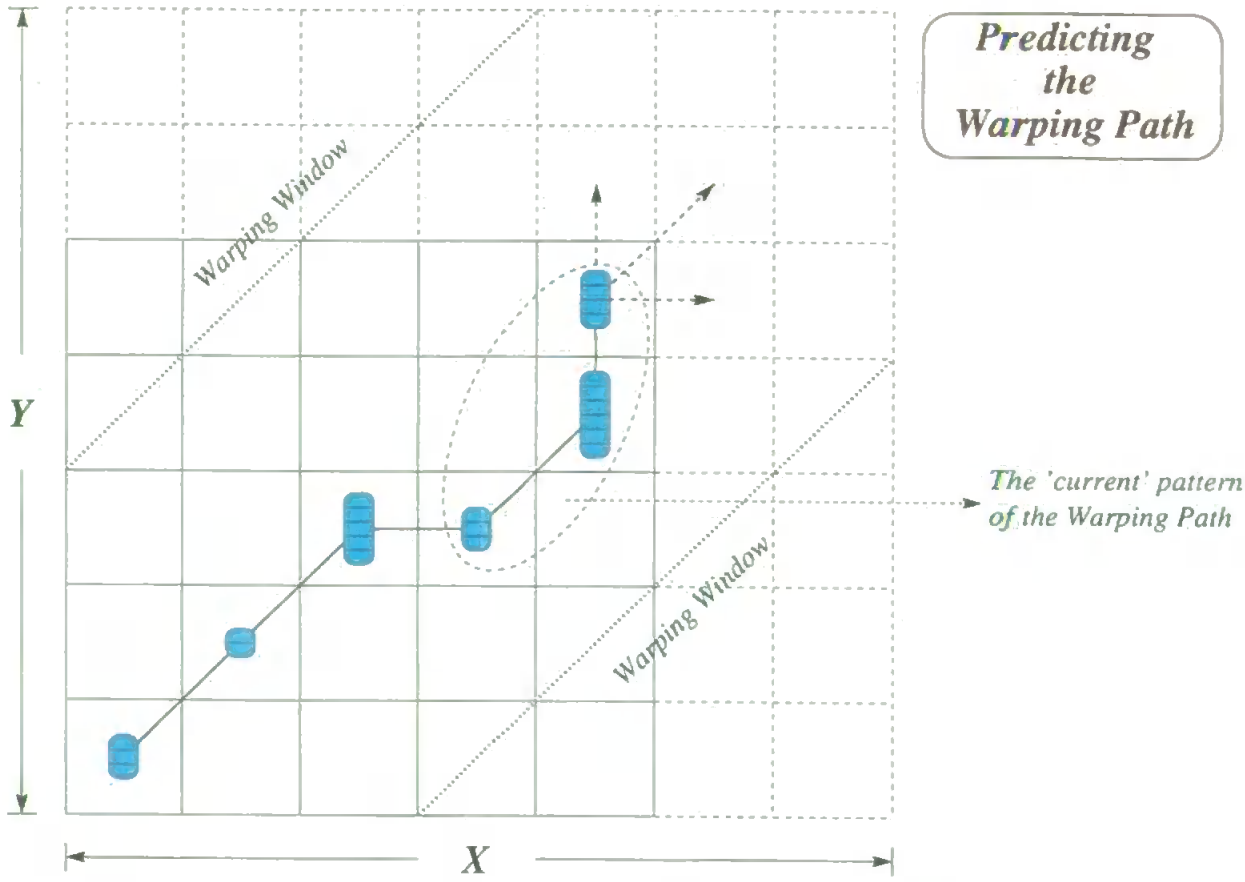


Figure 6.11: Graphical representation of the way that the warping path can be predicted. The small rectangles of different height represent the normalized cumulative distances which correspond to the warping path positions - cells - in the DTW-matrix. The dashed lines at the end of the warping path show the possible next step warping path movements.

coordinate series is required. These series are in ascending form and for warping path movement patterns to be revealed, we transform the series according to the following rule:

RULE: "*Write* **zero** *in the x-coordinates (y-coordinates) series for movements of the warping path along the same column (row) in the DTW-matrix and* **one** *for movements to successive columns (rows).*"

Having performed this transformation, the pattern matching prediction algorithm is applied to search for historical pattern movements of the warping path which are similar to

its "current" pattern movement. The length (number of warping path cells) of the recent pattern movement of the warping path is selected *a-priori*. Statistically derived techniques, like embedding dimension measures, which may identify patterns of non-linear deterministic behaviour within the warping path, may add some value to the prediction task by defining the length of the "current" warping path pattern. However, this matter is not investigated here.

In the next section, we present some results which have been obtained after applying the pattern matching prediction algorithm to predict the warping path generated for the British, French, German, Spanish, Italian and Greek stock market indices.

RESULTS In table 6.6, we present the performance of the pattern matching prediction algorithm in predicting the next day relative movement of the warping path. More specifically this involves the question of whether one series will delay against the other for the next time step or not, by having the co-integration relationship given. Speaking in terms of the time series warping path, we actually predict whether the warping path will move along the current column (series X will delay against Y, row (series Y will delay against X) or diagonally (synchronous time movement is more likely to occur) for the next time step. A random guess, to predict the next movement of the warping path within the DTW-matrix, would reveal a 33.33% opportunity of getting the correct movement. The pattern matching algorithm outperforms this almost every time. The average directional ability percentage of the system is greater than 38%. These results have been produced for the European DATASTREAM market indices over a ten years period (01.01.1990 - 26.04.2000). The prediction algorithm was tested on 50% of the warping path points.

The pattern matching algorithm, applied to warping path direction movement prediction seems to work better for longer time periods. As seen in table 6.6, the prediction accuracy is on average higher for the 10-year periods than for the 2-year ones. This maybe indicates that long warping paths support the pattern matching prediction algorithm.

The actual value prediction ability of the algorithm applied on the residual series is pre-

sented in table 6.7. As shown there, the root mean square prediction error (RMSE) does not exceed on average the value of 3.63% for the 2-year daily data and the value of 2.21% for the 10-year period. Once more, the prediction RMSE seems to be smaller for the 10-year data set.

Arbitrage is a trading behaviour, which mainly takes advantage of co-integrating relationships in order to buy cheap and sell expensive. In this context, we rely on the previous

Market Index	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Aver.	'90-2000
UK - FRANCE	38.69	41.16	39.38	41.54	42.31	40.62	41.54
UK - GERMANY	39.23	38.46	38.87	42.70	36.71	39.19	46.32
UK - SPAIN	43.67	33.72	41.09	36.04	41.16	39.14	39.67
UK - ITALY	39.44	33.58	31.79	45.64	39.62	38.01	39.35
UK - GREECE	46.74	39.85	38.39	42.31	37.54	40.97	43.11
FRANCE - GERMANY	41.54	39.23	40.51	50.77	42.31	42.87	48.29
FRANCE - SPAIN	37.69	37.69	44.03	37.55	40.77	39.55	39.01
FRANCE - ITALY	41.93	37.74	38.71	41.93	32.05	38.47	40.95
FRANCE - GREECE	34.47	43.62	40.00	32.12	43.08	38.66	43.74
GERMANY - SPAIN	42.69	45.77	45.96	42.32	43.56	44.06	45.21
GERMANY - ITALY	36.54	36.54	41.09	41.54	41.34	39.41	41.12
GERMANY - GREECE	43.24	35.91	40.05	44.23	40.95	40.88	42.32
SPAIN - ITALY	36.54	43.08	39.93	38.93	44.47	40.59	41.28
SPAIN - GREECE	42.60	37.84	42.25	33.36	38.06	38.82	39.57
ITALY - GREECE	42.53	40.77	39.93	36.54	40.39	40.03	39.85

Table 6.6: Results (directional ability (%)) after predicting the next step movement of the warping path using pattern matching. The warping path direction movement has been predicted for combinations of the six European DATASTREAM market indices. A random guess corresponds to 33.33%.

warping path prediction scheme to form a trading strategy which derives from statistical arbitrage.

Market Index	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Aver.	'90-2000
UK - FRANCE	3.97	2.51	4.37	3.05	3.59	3.50	2.21
UK - GERMANY	4.23	2.32	4.36	3.26	3.99	3.63	1.67
UK - SPAIN	4.11	2.14	3.92	3.03	3.75	3.39	1.63
UK - ITALY	4.18	2.32	3.91	3.62	3.86	3.59	1.73
UK - GREECE	3.81	2.35	3.73	2.26	3.59	2.39	1.65
FRANCE - GERMANY	3.66	3.10	3.74	3.29	2.55	3.27	1.19
FRANCE - SPAIN	3.86	2.85	3.81	3.42	2.55	3.30	1.04
FRANCE - ITALY	3.65	3.19	3.46	4.01	2.47	3.36	1.12
FRANCE - GREECE	3.51	3.69	3.41	3.52	2.14	3.25	1.18
GERMANY - SPAIN	3.07	2.10	4.95	3.24	3.51	3.37	1.43
GERMANY - ITALY	3.26	2.47	4.89	3.42	3.15	3.44	1.58
GERMANY - GREECE	3.24	2.85	5.01	3.26	3.26	3.52	1.46
SPAIN - ITALY	3.82	2.39	3.44	3.22	4.27	3.43	1.52
SPAIN - GREECE	3.66	2.65	3.59	2.95	4.21	3.41	1.56
ITALY - GREECE	2.68	3.83	3.00	3.85	3.32	3.34	1.46

Table 6.7: Results (Root Mean Square Prediction Error (%)) after predicting the next step cumulative distance of the warping path using pattern matching. The warping path cumulative distance value has been predicted for combinations of the six European DATASTREAM market indices.

6.2.5 Statistical Arbitrage - Evaluation

Generally, the Arbitrage Pricing Theory (APT) offers an alternative to the traditional asset pricing model in finance. Basically, arbitrage involves the law of one price for the same item. That is one item should not be sold at different prices. If this is not the case, arbitrage trading opportunities appear, through which arbitrageurs buy the item at low prices and simultaneously sell at higher ones. This kind of trading will continue until the different prices for one item become equal. Popular usage has expanded the meaning of the term "arbitrage" to include any activity which attempts to buy a relative underpriced item and sell a similar overpriced one, expecting to profit when the prices resume a more appropriate theoretical and/or historical relationship. This effect can be linked to some extent, to co-integration. Co-integration suggests that economic, financial, political and other factors cause the prices to move together in a long-term relationship and that the effect of "mispricing" will cause prices that diverge from the "fair" price to move back together. However, the main difference between co-integration and arbitrage relationships, as stated by BURGESS et al., (1996), [24], can be summarised as follows: "*co-integration is a statistical rather than a guaranteed relationship. The fact that prices have moved together in the past can suggest that they will continue to do so in future but cannot guarantee that this will in fact happen*". Because of that difference, BURGESS et al., (1996), [24] have considered the arbitrage opportunities that derive from the "*mispricing*" effect of the co-integration theory, as "**statistical arbitrage**", opposed to the classic arbitrage which involves no risk.

In section 6.2.3, we defined the equilibrium time warped co-integration relationship with a formula that is reproduced here:

$$y_t = \alpha + \beta \, \frac{x_t + x_{t-1} + \cdots + x_{t-k_t}}{k_t + 1} \quad . \tag{6.10}$$

The disequilibrium or 'mispricing error series of this relationship, can therefore be calculated as follows:

$$v_t = y_t - \alpha - \beta \, \frac{x_t + x_{t-1} + \cdots + x_{t-k_t}}{k_t + 1} \quad . \tag{6.11}$$

Following, the notation of BURGESS, (1997), [25], we expect the co-integration residual of equation 6.11 to be zero mean reverting and thus values of v_t that diverge from zero are considered as opportunities for statistical arbitrage. In this work, our expectation does not solely depend on mean reversion but also uses the disequilibrium residual prediction model (section 6.2.4). The decision to adopt a trading position at a particular moment using statistical arbitrage is taken when the prediction of the disequilibrium error tends to move towards zero. Additionally, the time index of the trading actions is decided by the predictive indication for the future movement of the warping path (see section 6.2.4). To formalise these arbitrage trading ideas using time warped co-integration, we write the following trading rule, which transforms the disequilibrium error predictions into trading positions:

- **if** $\hat{v}_{t+1}$ converge ($v_t > \hat{v}_{t+1}$), **then**
  - **if** $v_t < 0$, **then** *go long* (position = $+1$)
  - **if** $v_t > 0$, **then** *go short* (position = -1)
  - **if** $v_t = 0$, **then** *hold* (position = 0)
- **else** (i.e. $\hat{v}_{t+1}$ diverge ($v_t \leq \hat{v}_{t+1}$)), **then** no trading position is taken

v_t is the disequilibrium error (see equation 6.11) and $\hat{v}_{t+1}$ is the prediction of the disequilibrium error via the warping path prediction algorithm, which have been previously described. Furthermore, *going short* in the portfolio actually means selling the index which is taken as the reference index in the portfolio and buying the other one. Just the opposite happens when *going long*. The buy sell transactions may happen in different times. The closing arbitrage position may occur in the future. That time point is indicated by the prediction of the relative movement between the indices, i.e. the prediction of the warping path movement. The profit, therefore, is calculated as *return* multiplied

by *position*[9].

$$return = \frac{\Delta(y_t) - \Delta(\alpha + \beta x_t)}{y_t + \alpha + \beta x_t} \tag{6.12}$$

i.e.

$$return = \frac{change\ in\ combined\ portfolio}{absolute\ value\ of\ the\ two\ parts\ of\ the\ portfolio} \tag{6.13}$$

TRADING RESULTS Here we apply the *time-warped co-integration* framework together with the warping path prediction model on the same set of European stock market indices. Statistical arbitrage trading has been applied to market indices pairs whose residuals have t-statistic values more negative than -1.90 (see table 6.3)[10] for both the classical and the time-warped co-integration frameworks. Moreover, tests have been made separately for 2-year periods of the data as well as on the whole data set. For every test 40% of the data has been used as the time series historical part needed for the prediction model to perform. All profit measures are accompanied by the corresponding *Sharpe Ratio* (SR) risk metric. This is a measure of risk-adjusted return which represents the ratio of the annualised return divided by the standard deviation of the return. Trading costs are not taken into account, [25]. These results are summarised in tables 6.10, 6.9, 6.8, C.1, C.2, C.3, C.4, C.5, C.6. On average this shows that statistical arbitrage trading, based on time-warped co-integration, outperforms on the set of European DATASTREAM market indices the arbitrage trading strategy that derives from the classical co-integration theory for almost 80% of the time.

6.2.6 Conclusion

Seen in the time-warped co-integration study of this section, only the case of a pair of time series has been addressed. To extend this to N series, one can calculate the linear regression series of $N-1$ and search for co-integration between the regression series and the remaining one. However, a more general method involves the extension of the dynamic

[9]Private communication between the author and A.N. Burgess

[10]t-stat values close to the Dickey-Fuller critical value may lead to confusing results.

	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Aver.	'90-2000
SA	25.80	29.17	24.69	19.92	33.29	26.57	32.01
SA-DTW	41.49	38.49	30.30	41.71	65.66	43.53	42.67

Table 6.8: UK - FRANCE DATASTREAM market indices. SA and SA-DTW correspond to Statistical Arbitrage and Statistical Arbitrage using Dynamic Time Warping respectively. Profit is measured in number of basis points. No trading costs are included in the calculations.

time warping algorithm to more than two dimension, say N. Consider for example a 3-D optimal warping path in a cubic DTW-matrix. Such a path could be used for mapping three time series in order to retrieve time dependencies and to test for time-warped co-integration among them. Further increase in the dimension of the input space, therefore, may be viable. Although we believe, that there is much potential in investigating this idea, we leave it as part of our future work.

	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Aver.	'90-2000
SA	22.99	20.58	48.03	43.57	20.76	31.19	53.07
SA-DTW	31.74	44.42	50.09	15.92	43.45	37.12	78.62

Table 6.9: UK - GERMANY DATASTREAM market indices. SA and SA-DTW correspond to Statistical Arbitrage and Statistical Arbitrage using Dynamic Time Warping respectively. Profit is measured in number of basis points. No trading costs are included in the calculations.

	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Aver.	'90-2000
SA	48.72	53.84	46.52	3.44	46.62	39.83	26.55
SA-DTW	23.06	49.47	25.32	48.49	40.64	37.40	72.45

Table 6.10: FRANCE - GERMANY DATASTREAM market indices. SA and SA-DTW correspond to Statistical Arbitrage and Statistical Arbitrage using Dynamic Time Warping respectively. Profit is measured in number of basis points. No trading costs are included in the calculations.

6.3 Discussion and Outlook

In this chapter, we proposed two separate methods for trading multivariate economic time series. The first method is based on predictions made by a generalized version of the pattern matching algorithm described in chapter 5. The second method is based on a novel way of seeking for co-integrating long run relationships between two economic time series by considering local time delays or shifts between them.

The main hypothesis behind the pattern matching approach is that multivariate data sets add not only quantitative but also qualitative predictive information to the forecasting task. A necessary dimensionality reduction mechanism based on principal component analysis and correlation measures, distinguishes from a given data set those series that are highly correlated to its principal components. The time series selected improve the overall performance in terms of prediction and computational time. Testing the pattern matching approach on a set of European stock market indices has revealed profitability and risk reduction. Although, the inferences shown for the European market indices data set are statistically significant, we cannot say that the pattern matching system will universally perform the same way. Further tests using other data sets must be performed to support the forecasting and trading stability of the multivariate pattern matching system.

The co-integration framework we proposed in the second part of this chapter, is derived from a mechanism based on the dynamic time warping algorithm to identify local time relations between two economic time series. As a consequence of the dynamic time warping algorithm, we called this long run time series relationships framework, *time-warped co-integration.* In this part of the chapter, we introduced the concept of time-warped co-integration and showed how local time delays or shifts between two economic time series can be seen as variable time lags in a co-integrating framework. Following common practice found in the literature, we evaluated the time-warped co-integration framework through statistical arbitrage trading and compared the results against those obtained using the classic co-integration practise. The simulations on the European stock market indices set have shown promising results. Almost 80% of the time, the time warped version of co-integration gave higher profits or lower losses compared to the classic version of co-integration. The *Sharpe Ratio* risk measures also appear to drop correspondingly for the data set used. However, tests on different data sets, like macroeconomic indicators will also be interesting. An important point that will be investigated in the future research is the way that the DTW local time transformation of the series affects their level of integration, i.e. their stationarity properties. A point that also has to be stressed, is that the case examined here involves only pairs of financial time series. This has been deliberately done in order to examine the system in its simplest form. It is viable though, to expand the concept of time-warped co-integration to more than two time series. In that case, a n-dimensional version of the dynamic time warping algorithm may be implemented and tested. In this case the corresponding optimal warping path calculated within the n-dimensional DTW-matrix would identify local time relationships among a set of n time series. Therefore, a portfolio of n financial time series may be tested for *multiple time-warped co-integration.* However, we leave the systematic examination and development of this idea as part of our future research.

Chapter 7

Discussion and Outlook

The Truth Is Out There.
I want to Believe.

Slogans from The X-files.

Forecasting and trading on financial markets is a hard task, since under the very broad laws of demand and supply there is no specific mechanism that drives the market prices. A large number of people and organizations participate simultaneously in the markets and they constantly change the "rules of the game". How does this situation reflect on the financial time series? Volatile financial time series that sometimes seem to be generated from random processes, efficient markets, uncertainty in forecasting and risk and disappointment in trading could be offered as answers to the question above. Computational intelligence systems may, however, offer an oasis in the financial markets desert by continuously keeping an eye on the financial facts and alerting users to financial opportunities or dangers. The aim of this thesis was to add a small contribution to this context.

Three approaches to financial analysis are proposed in this thesis. A topological neural network, the Growing Neural Gas, that controls its architecture complexity according to the complexity of the problem it has to solve was applied on financial market indices

(chapter 3). The '*growing*' attribute of the network's architecture plus its ability to follow non-stationary distributions, according to tests performed on classification problems, was the motivation for applying Growing Neural Gas to financial forecasting and trading. To my knowledge this network has never been applied in this domain before.

The aim of the core part of this thesis was to build a computational intelligence pattern matching and recognition system that operates as an '*electronic eye*' over financial time series. Its purpose was to isolate the 'current' situation of the market, as derived from financial data series, and to draw predictive conclusions from historical patterns similar to the 'current' situation. In my opinion, financial pattern matching and recognition would be adequate for practitioners if it met the following objectives: operational simplicity, robustness, autonomy, robustness to noise and minor data changes and computational efficiency. No computational system in the financial literature satisfies all these requirements. Neural networks, for instance, are dependent on *a-priori* expert tuning and they are not famous for their operational simplicity. The systems proposed in chapters 4 and 5 owe their success to the flexible pattern matching algorithm that incorporates changes in the momentum of the financial data series' evolution. Time series piecewise linear segment representations, dynamic time warping, graph pattern matching and embedding dimension statistical measures have all contributed to build an autonomous and robust computationally efficient system that is based on the simple hypothesis that similarity with the past is essentially prediction.

In the last chapter (chapter 6) of this work, multivariate aspects of the pattern matching system have been addressed. A novel co-integration scheme, based on possible local time relations between two time series, is proposed. The hypothesis from which this research derived, was that local time shifts, delays, shrinks or stretches, if identified, could help to reveal co-integrating movement between two time series. The Dynamic Time Warping algorithm was used to identify these local time relations and stationarity statistical measures have supported this idea. The overall co-integrating scheme has been tested for arbitrage trading. It has been shown that statistical arbitrage trading based on time-warped co-integration outperforms the one that derives from the classical co-integration

theory.

The most useful way of developing financial forecasting systems for decision support or trading would be to develop them for real time use. Given that the exponential increase in processing power of standard computers is on our side, our future work should study computational finance applications in real time.

Appendix A

Technical Analysis Price Oscillators

Information is pretty thin stuff,
unless mixed with experience.

Clarence Day, 1920.

One of the most popular indicator classes in technical analysis (TA) theory is the one loosely referred to as oscillators. Among others, the relative strength index (RSI), the price oscillator (PrOsc), the moving average convergence-divergence (MACD) indicator and the rate of change (ROC) are some known representatives of TA oscillators. Usually oscillators are used by chartists as countertrend indicators. That is they are used to identify short-term price reversal points rather than long-term ones. Table A.1 shows the main attributes of the oscillator indicators as retrieved from the TA literature ([2, 121, 142, 143]).

	Oscillators:
1.	are useful in trending markets.
2.	alert traders to short market extremes - **overbought** or **oversold** conditions.
3.	warn traders when the trend is loosing momentum.
4.	signal when a trend is near completion by displaying certain divergences.
5.	are misleading at the beginning of important moves.
6.	are extremely valuable towards the end of trend movements.

Table A.1: Main technical analysis attributes.

A.1 Price Oscillator

A price oscillator (PrOsc) is simply the series that derives after differencing two moving averages of a financial time series. The difference between the moving averages can be expressed in either points or percentages[1]. A longer-term moving average that picks the broad changes in the time series, is subtracted from a shorter-term one that is responsible for identifying the short-term trend reversals. Such an oscillator is noted as **PrOscA − B**, where **A** represents an A-days moving average and **B** a B-days one ($A < B$).

$$PrOscA - B = (A - daysMA) - (B - daysMA) \tag{A.1}$$

or in percentages

$$PrOscA - B = \frac{(A - daysMA) - (B - daysMA)}{(A - daysMA)} * 100 \tag{A.2}$$

Figure A.1 shows the curve of the PrOsc 10-30 as applied on the Amazon security drawn between May 1999 and Feb 2000. The moving averages are drawn over the actual time series (green and red lines) and on the top both the absolute and the percentage price oscillators are depicted (black and blue lines). As seen in the figure the oscillator

[1]The Price Oscillator is almost identical to the MACD, except that the Price Oscillator can use any two user-specified moving averages. (The MACD always uses 12 and 26-day moving averages, and always expresses the difference in points.)

Figure A.1: The PrOsc 10-30 (both absolute and percentage mode) applied on the Amazon.com security. The green and the red lines represent the 10- and 30-days moving averages. The blue and the black curves on the top of the figure are the percentage and absolute price oscillators respectively. (This figure obtained from StockCharts.com Inc.)

curves fluctuate on the zero equilibrium lines showing when the difference between the short-term and the long-term averages is positive or vice versa. The crossover points, the points where the price oscillator crosses the equilibrium line, produce trading *buy* and *sell* signals. Everytime that the price oscillator curve crosses the equillibrium line from below a *buy* signal is considered. *Sell* signals are generated when the equillibrium line is crossed from above. However, taking into account that the PrOsc trading signals are generated delayed against the actual trend reversals of the security time series, the PrOsc's are used in combination with their own moving averages[2]. This is done by calculating the C-days moving average of the price oscillator curve. The crossover points between the oscillator curve and its moving average generate the trading signals similar to the way that has

[2]See [2, 121, 142, 143] for a detailed analysis on price oscillators.

been described above. Crossovers of the oscillator curve from below represent *buy* signals while others from above indicate *sell* actions. The notation of a price oscillator is therefore, summarised as $PrOscA - B/C$ where A, B, C are the moving average parameters needed to construct the price oscillator indicator in order to generate trading signals.

In conclusion ([121, 142, 143, 53]):

1. PrOsc can be used by everyone because their implementation and evaluation on particular securities are very simple and do not require any specialized software.

2. Usage of PrOsc has proven to be very useful and reliable for supporting trading decisions, particularly when trading decisions are difficult to make.

3. When PrOsc are employed for trading, users must rely on their effectiveness and avoid 'second thoughts'.

4. PrOsc based trading prevents everyday trading and thus avoids extensive transaction costs.

5. PrOsc are very reliable in clear ascending or descending markets and they guarantee that in such markets price oscillator driven investors will be "in" the market in ascending periods and out of it is descending ones.

Appendix B

Further Results on Pattern Matching

On the following pages some more results (forecasting and trading) of the univariate pattern matching architecture are presented. These results concern the FRANCE-DS-MK, GERMANY-DS-MK, SPAIN-DS-MK, ITALY-DS-MK, GREECE-DS-MK DATASTREAM market indices. The following tables are presented in the same way as those in chapter 4 (tables 4.1, 4.2).

FRANCE-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
MED	61.9	58.4	60	61	57.6	**59.8**	**58.1**
PrOsc	57.3	54.9	55.5	56.5	59.1	**56.7**	**59.8**
LSR	52.3	50.8	64.8	55.1	62.7	**57.1**	**56.9**
Forecasting Sentences (FS)							
MED	56.7	52.6	63.9	57.3	53.5	**56.8**	**54.3**
PrOsc	59.2	52	59.9	53.1	45.1	**53.9**	**52.1**
LSR	59.3	55.4	56.3	53	46.7	**54.1**	**51.1**
GNG Forecasting (GNG-F)							

Table B.1: Testing the forecasting directional ability of the pattern matching system. The FRANCE-DS-MK index case. MED, PrOsc and LSR stand for Minimum Embedding Dimension, Price Oscillator and Linear Segment Representation respectively. These correspond to the three methods for selecting the query pattern.

GERMANY-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
MED	56.9	49.4	58.8	62.7	61.4	**57.8**	**55.6**
PrOsc	57.4	51.2	59.1	56.9	58.4	**56.6**	**58.3**
LSR	55.3	46.9	57.7	62.3	56.9	**55.8**	**58.5**
Forecasting Sentences (FS)							
MED	60.7	53.6	61.1	46.8	50.9	**54.6**	**50.9**
PrOsc	54.9	54.5	63.0	49.0	52.8	**54.8**	**50.1**
LSR	54.4	44.8	56.5	47.6	54.5	**51.6**	**49.1**
GNG Forecasting (GNG-F)							

Table B.2: Testing the forecasting directional ability of the pattern matching system. The GERMANY-DS-MK index case. MED, PrOsc and LSR stand for Minimum Embedding Dimension, Price Oscillator and Linear Segment Representation respectively. These correspond to the three methods for selecting the query pattern.

SPAIN-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
MED	57.5	53.5	64.7	59.5	61.5	**59.3**	**58.2**
PrOsc	58.9	62.6	60.8	60.8	51.1	**58.8**	**57.8**
LSR	55.8	58.8	55.1	53.9	55.4	**55.8**	**58.4**
Forecasting Sentences (FS)							
MED	56.1	51.7	55.2	46.5	54.6	**52.8**	**48.7**
PrOsc	52.4	56.6	61.3	52.6	58.4	**56.3**	**52.3**
LSR	55.4	47.2	53.6	48.7	55.2	**52.0**	**51.2**
GNG Forecasting (GNG-F)							

Table B.3: Testing the forecasting directional ability of the pattern matching system. The SPAIN-DS-MK index case. MED, PrOsc and LSR stand for Minimum Embedding Dimension, Price Oscillator and Linear Segment Representation respectively. These correspond to the three methods for selecting the query pattern.

ITALY-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
MED	55.8	61.5	59.5	57.9	56.9	**58.3**	**57.2**
PrOsc	54.2	63.7	54.4	61.9	61.6	**59.2**	**57.3**
LSR	58.8	58.3	59.3	60.6	52.6	**57.9**	**58.8**
Forecasting Sentences (FS)							
MED	51.8	53.8	55.4	53.6	59.5	**54.8**	**54.9**
PrOsc	53.6	52.5	52.5	50.8	58.9	**53.4**	**52.8**
LSR	60.7	48.6	58.2	52.4	53.9	**54.8**	**54.1**
GNG Forecasting (GNG-F)							

Table B.4: Testing the forecasting directional ability of the pattern matching system. The ITALY-DS-MK index case. MED, PrOsc and LSR stand for Minimum Embedding Dimension, Price Oscillator and Linear Segment Representation respectively. These correspond to the three methods for selecting the query pattern.

GREECE-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
MED	61.0	60.8	58.1	58.9	61.8	**60.1**	**60.2**
PrOsc	60.2	58.7	62.2	56.9	55.8	**58.8**	**60.1**
LSR	56.4	55.7	62.0	56.4	57.8	**57.7**	**59.5**
Forecasting Sentences (FS)							
MED	59.5	62.9	58.5	57.3	57.6	**59.2**	**56.2**
PrOsc	58.7	58.1	61.1	54.9	54.7	**57.5**	**54.5**
LSR	52.2	50.1	52.8	61.5	59.2	**55.2**	**55.7**
GNG Forecasting (GNG-F)							

Table B.5: Testing the forecasting directional ability of the pattern matching system. The GREECE-DS-MK index case. MED, PrOsc and LSR stand for Minimum Embedding Dimension, Price Oscillator and Linear Segment Representation respectively. These correspond to the three methods for selecting the query pattern.

FRANCE-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
MED	13.94	1.1	17.55	9.29	4.83	**9.34**	**7.98**
	(1.59)	(0.03)	(2.26)	(0.91)	(0.15)	(0.99)	(0.28)
PrOsc	8.46	3.95	10.96	6.83	7.63	**7.57**	**7.70**
	(0.79)	(0.11)	(1.22)	(0.30)	(0.39)	(0.56)	(0.26)
LSR	8.61	-8.94	16.17	9.08	8.18	**6.62**	**8.87**
	(0.81)	(-1.49)	(2.09)	(0.87)	(0.60)	(0.58)	(0.57)
Forecasting Sentences (FS)							
MED	11.94	-4.50	18.18	14.10	2.89	**8.52**	**9.11**
	(1.22)	(-0.58)	(2.42)	(1.45)	(0.05)	(0.91)	(0.78)
PrOsc	13.12	-1.92	10.01	1.66	-5.86	**3.40**	**6.17**
	(1.37)	(-0.22)	(1.00)	(0.14)	(-1.25)	(0.21)	(0.22)
LSR	14.80	-0.25	13.26	4.62	-4.19	**5.65**	**6.25**
	(1.59)	(-0.11)	(1.52)	(0.42)	(-0.93)	(0.50)	(0.24)
GNG Forecasting (GNG-F)							

Table B.6: Testing the trading ability of the pattern matching system. The CAC 40 (FRANCE) index case. The values in quotes are the corresponding Sharpe Ratio (SR) measures.

GERMANY-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
MED	9.90	2.62	16.46	8.68	6.65	**8.86**	**7.61**
	(1.17)	(0.34)	(2.54)	(1.93)	(0.83)	(1.36)	(2.58)
PrOsc	10.82	-3.23	8.79	12.09	8.89	**7.47**	**7.83**
	(1.16)	(-0.64)	(0.96)	(1.26)	(1.04)	(0.76)	(2.80)
LSR	9.12	4.88	9.79	3.71	1.08	**5.72**	**8.18**
	(1.15)	(0.34)	(1.35)	(0.87)	(0.17)	(0.78)	(2.92)
Forecasting Sentences (FS)							
MED	9.83	7.22	14.34	3.95	-1.76	**6.72**	**6.52**
	(2.21)	(0.98)	(1.09)	(0.44)	(-0.14)	(0.92)	(1.64)
PrOsc	11.35	0.74	13.47	6.58	-6.02	**5.22**	**7.34**
	(3.48)	(0.14)	(1.14)	(0.81)	(-0.74)	(0.97)	(1.77)
LSR	10.09	-3.21	-2.48	7.16	3.56	**3.02**	**6.58**
	(1.40)	(-0.58)	(-0.19)	(0.93)	(0.32)	(0.38)	(1.44)
GNG Forecasting (GNG-F)							

Table B.7: Testing the trading ability of the pattern matching system. The GERMANY-DS-MK index case. The values in quotes are the corresponding Sharpe Ratio (SR) measures.

SPAIN-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
MED	14.74	2.68	15.11	7.77	14.38	**10.94**	**9.86**
	(1.55)	(0.43)	(2.45)	(0.54)	(2.14)	(1.42)	(1.53)
PrOsc	11.50	4.28	6.24	6.24	4.07	**6.47**	**9.23**
	(1.18)	(1.26)	(1.11)	(1.32)	(0.79)	(1.13)	(3.63)
LSR	10.42	4.50	8.66	1.38	-0.44	**4.90**	**8.02**
	(1.04)	(1.26)	(1.30)	(0.13)	(-0.48)	(0.65)	(3.56)
Forecasting Sentences (FS)							
MED	14.27	3.32	7.74	6.45	5.68	**7.49**	**6.71**
	(1.49)	(0.51)	(0.96)	(0.99)	(0.77)	(0.94)	(1.44)
PrOsc	20.60	4.30	10.72	4.27	12.15	**10.41**	**7.62**
	(1.89)	(0.66)	(1.09)	(0.69)	(1.28)	(1.12)	(1.31)
LSR	13.97	0.48	-1.19	4.21	9.64	**5.42**	**8.72**
	(1.72)	(0.16)	(-0.16)	(0.69)	(1.19)	(0.72)	(2.31)
GNG Forecasting (GNG-F)							

Table B.8: Testing the trading ability of the pattern matching system. The SPAIN-DS-MK index case. The values in quotes are the corresponding Sharpe Ratio (SR) measures.

ITALY-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
MED	11.94	15.75	15.39	5.96	2.14	**10.24**	**8.63**
	(1.08)	(2.13)	(3.75)	(0.80)	(0.30)	(1.61)	(2.89)
PrOsc	12.47	9.50	16.31	5.25	2.50	**9.21**	**7.31**
	(1.72)	(1.33)	(2.19)	(1.02)	(0.32)	(1.32)	(2.72)
LSR	10.81	9.97	11.92	5.11	9.91	**9.54**	**8.71**
	(1.35)	(1.04)	(1.59)	(0.81)	(1.00)	(1.16)	(2.63)
Forecasting Sentences (FS)							
MED	9.02	10.05	8.45	-1.23	6.97	**6.65**	**5.66**
	(2.62)	(1.12)	(2.04)	(-0.37)	(0.93)	(1.27)	(1.19)
PrOsc	16.04	9.89	13.60	0.30	4.53	**8.87**	**8.91**
	(1.86)	(1.38)	(1.88)	(0.05)	(0.63)	(1.03)	(2.37)
LSR	13.13	0.65	9.48	3.32	1.76	**5.67**	**9.11**
	(2.31)	(0.09)	(2.17)	(0.38)	(0.19)	(1.03)	(1.95)
GNG Forecasting (GNG-F)							

Table B.9: Testing the trading ability of the pattern matching system. The ITALY-DS-MK index case. The values in quotes are the corresponding Sharpe Ratio (SR) measures.

GREECE-DS-MK

	1990-92	1992-94	1994-96	1996-98	1998-2000	Average	1990-2000
MED	15.02	14.81	6.67	11.70	9.09	**11.46**	**8.69**
	(2.18)	(1.49)	(0.82)	(1.30)	(1.52)	(1.46)	(2.64)
PrOsc	20.27	8.89	5.78	11.24	8.07	**10.85**	**10.13**
	(2.87)	(0.88)	(0.72)	(1.14)	(1.13)	(1.35)	(1.04)
LSR	14.72	6.63	8.60	8.88	9.65	**9.70**	**11.06**
	(2.09)	(0.72)	(0.98)	(1.74)	(1.05)	(1.32)	(1.31)
Forecasting Sentences (FS)							
MED	18.13	7.66	6.36	7.19	5.51	**8.97**	**7.75**
	(2.30)	(0.63)	(0.74)	(1.36)	(0.69)	(1.14)	(1.39)
PrOsc	17.27	13.28	4.69	8.32	9.47	**10.61**	**8.60**
	(2.40)	(1.23)	(0.45)	(1.14)	(1.07)	(1.26)	(1.72)
LSR	13.75	-5.08	-2.16	12.68	6.76	**5.19**	**8.74**
	(1.75)	(-0.58)	(-0.45)	(1.36)	(0.82)	(0.58)	(1.87)
GNG Forecasting (GNG-F)							

Table B.10: Testing the trading ability of the pattern matching system. The GREECE-DS-MK index case. The values in quotes are the corresponding Sharpe Ratio (SR) measures.

Appendix C

Further Statistical Arbitrage Results

On the following pages some more results concerning the comparison of the co-integration and time-warped co-integration statistical arbitrage trading are presented. The tables in this appendix are presented in the same way as those in section 6.2.5 (tables 6.8, 6.9, 6.10).

	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Aver.	'90-2000
SA	10.18	42.64	14.53	19.31	64.88	30.31	30.20
SA-DTW	20.77	30.84	66.58	20.49	24.15	32.57	70.95

Table C.1: UK - SPAIN DATASTREAM market indices. SA and SA-DTW correspond to Statistical Arbitrage and Statistical Arbitrage using Dynamic Time Warping respectively. Profit is measured in number of basis points. No trading costs are included in the calculations.

	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Aver.	'90-2000
SA	56.35	10.58	21.75	65.05	48.75	40.50	3.10
SA-DTW	60.25	31.51	31.80	39.03	64.15	45.35	14.25

Table C.2: UK - GREECE DATASTREAM market indices. SA and SA-DTW correspond to Statistical Arbitrage and Statistical Arbitrage using Dynamic Time Warping respectively. Profit is measured in number of basis points. No trading costs are included in the calculations.

	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Aver.	'90-2000
SA	61.80	21.02	14.94	27.70	57.22	36.54	60.74
SA-DTW	24.93	33.79	46.29	19.18	34.59	31.76	46.76

Table C.3: GERMANY - SPAIN DATASTREAM market indices. SA and SA-DTW correspond to Statistical Arbitrage and Statistical Arbitrage using Dynamic Time Warping respectively. Profit is measured in number of basis points. No trading costs are included in the calculations.

	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Aver.	'90-2000
SA	54.58	17.23	10.61	61.77	73.83	43.60	51.43
SA-DTW	63.35	59.17	27.55	43.38	33.35	45.36	67.43

Table C.4: GERMANY - ITALY DATASTREAM market indices. SA and SA-DTW correspond to Statistical Arbitrage and Statistical Arbitrage using Dynamic Time Warping respectively. Profit is measured in number of basis points. No trading costs are included in the calculations.

	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Aver.	'90-2000
SA	2.82	22.49	17.66	16.81	41.76	20.31	44.43
SA-DTW	34.66	28.26	45.42	37.51	18.32	32.83	43.64

Table C.5: SPAIN - ITALY DATASTREAM market indices. SA and SA-DTW correspond to Statistical Arbitrage and Statistical Arbitrage using Dynamic Time Warping respectively. Profit is measured in number of basis points. No trading costs are included in the calculations.

	'90-'92	'92-'94	'94-'96	'96-'98	'98-2000	Aver.	'90-2000
SA	10.50	68.85	11.61	17.80	4.25	22.60	34.12
SA-DTW	30.92	28.92	57.44	21.78	73.66	42.54	42.79

Table C.6: SPAIN - GREECE DATASTREAM market indices. SA and SA-DTW correspond to Statistical Arbitrage and Statistical Arbitrage using Dynamic Time Warping respectively. Profit is measured in number of basis points. No trading costs are included in the calculations.

Appendix D

PROGNOSIS

In this appendix, some snapshots of the financial time series simulator PROGNOSIS are given. Figure D.1, on the next page, is a snapshot of the PROGNOSIS simulator (version 1.1a) that performs the pattern matching algorithm that has been described in chapter 4. The next snapshot (figure D.2) shows the simulator that performs segmented pattern matching using dynamic time warping (chapter 5). The last snapshot that is given in figure D.3 depicts the PROGNOSIS simulator that performs multivariate pattern matching analysis and time-warped co-integration (chapter 6).

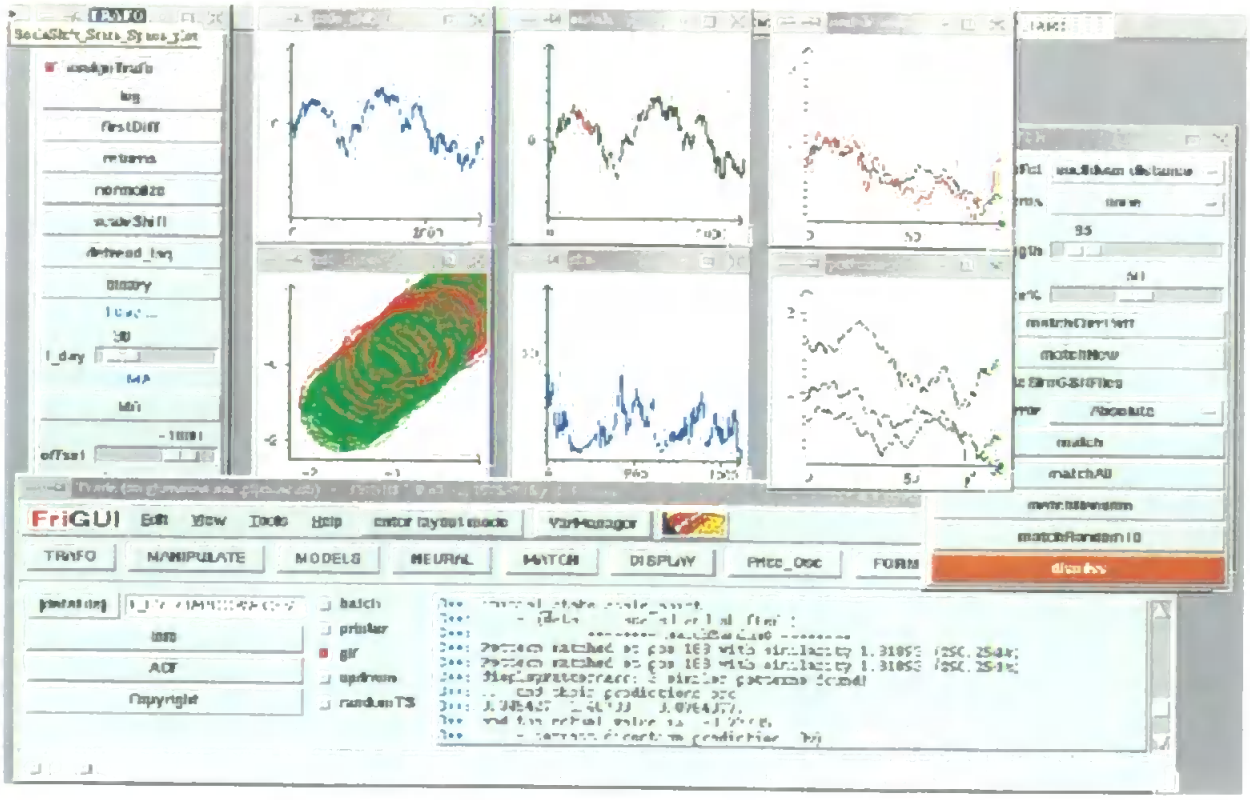


Figure D.1: Snapshot of the PROGNOSIS simulator (v1.1a).

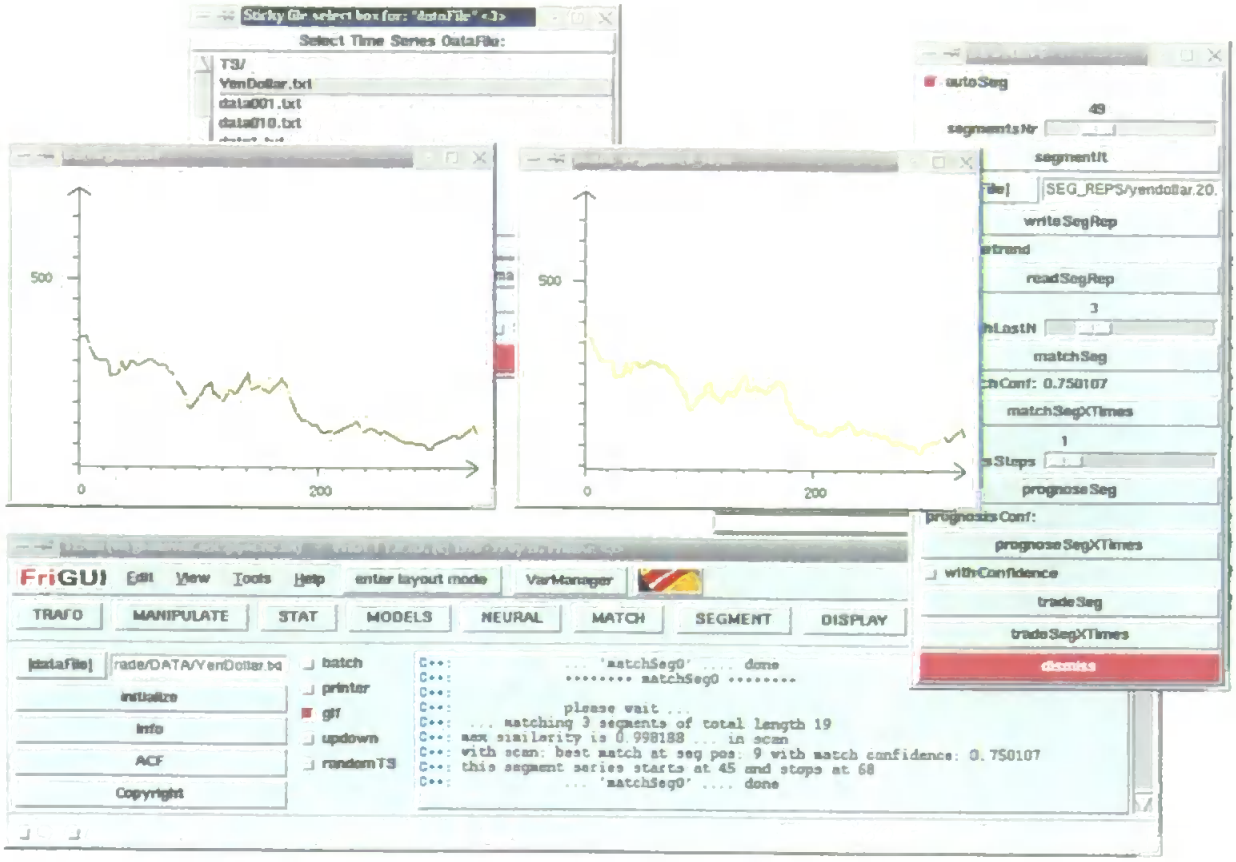


Figure D.2: Snapshot of the PROGNOSIS simulator (v1.2a).

Figure D.3: Snapshot of the PROGNOSIS simulator (v1.3a).

Appendix E

Abbreviations

AFDSS	Automated Forecasting Decision Support System
AMC	Average Matching Confidence
ANN	Artificial Neural Network
APC	Average Prediction Confidence
APT	Arbitrage Pricing Theory
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
CD	Correlation Dimension
CHL	Competitive Hebbian Learning
CI	Computational Intelligence
CP	Cumulative Profit
DA	Directional Ability
DFT	Discrete Fourier Transform
DTW	Dynamic Time Warping
ECM	Error Correction Model
ED	Embedding Dimension
EGM	Elastic Graph Matching
EMH	Efficient Market Hypothesis
ESD	Embedding Segment Dimension

FS	Forecasting Sentences
GA	Genetic Algorithm
GNG	Growing Neural Gas
ICA	Independent Component Analysis
IID	Independent and Identically Distributed
LSR	Linear Segment Representation
MA	Moving Average
MACD	Moving Average Convergence Divergence
MED	Minimum Embedding Dimension
MESD	Minimum Embedding Segment Dimension
MFS	Multiple Feature Sets
MLP	Multi Layer Perceptron
NG	Neural Gas
NT	Number of Transactions
OLS	Ordinary Least Squares
PCA	Principal Component Analysis
PrOsc	Price Oscillator
QR	Query Pattern
RBF	Radial Basis Functions
RMSE	Root Mean Square Error
ROC	Rate Of Change
RSI	Relative Strength Index
SOM	Self Organising Map
SR	Sharpe Ratio
StOsc	Stochastic Oscillator
SVD	Singular Value Decomposition
TA	Technical Analysis
UCM	Unobservable Components Models

Appendix F

Curriculum Vitae

Name	Georgios N. Banavas
Date of birth	30. April 1972
Place of birth	Thessaloniki, GREECE
Marital Status	Single
Address	7 Kirkby Place, PL4 8AA, Plymouth, UK
Telephone	+44 (0)40 / 3781852
Email	Georgios.Banavas@soc.plym.ac.uk

Education

07/98 – 09/00	Centre for Neural and Adaptive Systems, University of Plymouth, UK
01/97 – 06/98	Institut für Neuroinformatik, Ruhr-Universität Bochum, GERMANY
08/96	Mittelstufe III, Goethe Institut Thessaloniki, GREECE
04/96	PNdS, certificate in German, Ruhr Univeristät Bochum, GERMANY
09/90 – 03/96	Diplom in Electrical Engineering and Computer Science, Aristoteles University Thessaloniki, GREECE
10/90	1st certificate in English, University of Cambridge
09/78 – 06/90	School time, Gymnasium and Lyseum, N. Moudania, Chalkidiki, GREECE

Bibliography

[1] Abu-Mostafa, Y., (1993), 'Using Hints for Neural Network Training', *Invited Talk at the First International Workshop on Neural Networks in the Capital Markets, London.*

[2] Achelis, S. B., (2000), 'Technical Analysis from A to Z', *Equis International, Reuters Company.*

[3] Agrawal, R., Faloutsos, C, & Swami, A., (1993), 'Efficient Similarity Search In Sequence Databases', *Research Report, IBM Almaden Research Center, San Jose, California.*

[4] Agrawal, R., Lin, K.-I., Sawhney, H. S. & Shim, K., (1995), 'Fast Similarity Search in the Presence of Noise, Scaling and Translation in Time Series Databases', In *Proceedings of the 21st VLDB Conference, Zurich, Switzerland.*

[5] Andersen, T. G., (1992), 'A model of Return Volatility and Trading Volume', *Working paper, Northwestern University, Evanston, IL.*

[6] Balvers, R., Wu, Y. & Gilliland, E., (2000), 'Mean Reversion across National Stock Markets and Parametric Contrarian Investment Strategies', In *Journal of Finance, Vol. LV, No. 2.*

[7] Banavas, G., (1999), 'Univariate Time-Series Forecasting based on Complex Subsequence Pattern Matching', *Technical Report, Centre for Neural & Adaptive Systems, Plymouth.*

[8] Banavas, G., (1999), 'Robust Direction Movement Forecasting using a Pattern Matching Method applied on Univariate Financial Time Series', *ISF'99, Washington DC.*

[9] Banavas, G. N., Denham, S. & Denham, M. J., (2000), 'A Financial Stock Index Trend Prediction Approach with Temporal Considerations', *Computational Finance/Forecasting Financial Markets 2000, LBS, London.*

[10] Banavas, G. N., Denham, S. & Denham, M. J., (2000), 'Fast non-linear deterministic forecasting of segmented stock indices using pattern matching and embedding techniques.', *6^{t}h International Conference on Computing in Economics and Finance, CEF'2000, Pompeu Fabra, Barcelona.*

[11] Banavas, G. N. et al., (2000), 'Intelligent monitoring and communication system', In *Proceedings of SPIE, Vol. 3995: Nondestructive Evaluation of Highways, Utilities, and Pipelines IV, Newport Beach, California.*

[12] Beckmann, N., Kriegel, H.-P., Schneider, R., & Seeger, B., (1990), 'The r^*-tree: an efficient and robust access method for points and rectangles', In *ACM SIGMOD, pp. 322-331.*

[13] Bergio, Y., (1996), 'Training a Neural Network with a Financial Criterion Rather than a Prediction Criterion', In *Proceedings of the Neural Networks in the Capital Markets (NNCM'96).*

[14] Berndt, D. & Clifford, J., (1994), ' and J. Clifford Using Dynamic Time Warping to find patterns in time-series', In *AAAI94 workshop of knowledge discovery in databases.*

[15] Bessembinder, H. & Kaufman, H., (1997), 'A comparison of Trade Execution Costs for NYSE and Nasdaq-Listed Stocks', In *Journal of Financial and Quantitative Analysis, vol.32, No.3, (September): 287-310.*

[16] Bessembinder, H. & Kaufman, H., (1998), 'Trading Costs and Volatility for Technology Stocks', In *Financial Analysts Journal (September): 64-71.*

[17] Bishop, C. M., (1996), 'Neural Networks for Pattern Recognition', *Clareton Press, Oxford.*

[18] Blasco, N. & Santamaria, R., (1996), 'Testing Memory Patterns in the Spanish Stock Market', In *Applied Financial Economics, 6, pp. 401-411.*

[19] Box & Jenkins, (1976), 'Time Series Analysis: forecasting and control', *Revised Edition, eds. Holden Day..*

[20] Brock, W., Dechert, W., and Scheinkman, J., (1987), 'A Test for Independence Based On the Correlation Dimension', *University of Wisconsin, Madison, University of Houston, and University of Chicago.*

[21] Brock, W., Hsieh, D., and LeBaron, B., (1991), 'Nonlinear Dynamics, Chaos and Instability: Statistical Theory and Economic Evidence', *The MIT Press, Cambridge, Massachutets, London, England.*

[22] Brock, W. A., Lakonishok, J. & LeBaron, B., (1992), 'Simple Technical Trading Rules And The Stochastic Properties of Stock Returns', In *Journal of Finance, XLVII: 5, 1731-64.*

[23] Broomhead, D. S. and King, G. P., (1986), 'Extracting Qualitative Dynamics from Experimental Data', In *Physica D 20, pp. 217-236.*

[24] Burgess, A. N., & Refenes, A. N., (1996), 'Modelling nonlinear cointegration in international equity index futures', In *Neural Networks in Financial Engineering, (Singapore, World Scientific).*

[25] Burgess, A. N., (1997), 'Asset allocation across european equity indices using a portfolio of dynamic cointegration models', In *Decision Technologies in Financial Engineering, eds.*

Weigend et al., World Scientific.

[26] Burgess, A. N., (2000), 'Exploiting computational intelligence techniques within a framework for statistical arbitrage', In *Computational Finance/Forecasting Financial Markets 2000, LBS, London.*

[27] Cambell, J. Y. & Hentschel, L., (1993), 'No news is good news: An assymetric model of changing volatility in stock returns', In *Journal of Financial Economics.*

[28] Cambell, J. Y., Lo, A. W. & MacKinlay, A. C., (1997), 'The econometrics of the financial markets', *Princeton University Press, Princeton N.J..*

[29] Cao, L., Hong, Y., Fang, H. & He, G., (1995), 'Predicting Chaotic Time Series with Wavelet Networks', In *Physica D 85, pp. 225-238.*

[30] Cao, L., (1997), 'Practical Method for Determing the Minimum Embedding Dimension of a Scalar Time Series', In *Physica D 110, pp. 43-50.*

[31] Cao, L., (1998), 'Determing the Minimum Embedding Dimension of Input-Output Time Series Data', In *International Journal of Bifurcation and Chaos, Vol. 8, No. 7, pp. 1491-1504.*

[32] Cao, L., Mees, A. & Judd, K., (1998), 'Dynamics from Multivariate Time Series', In *Physica D, 121, pp. 75-88.*

[33] Cao, L. & Soofi, A. S., (1999), 'Nonlinear deterministic forecasting of daily dollar exchange rates', In *International Journal of Forecasting, Vol. 15, pp. 421-30.*

[34] Cecen, A. & Erkal, C., (1996), 'Distinguishing between stochastic and deterministic behavior in high frequency foreign exchange rate returns: Can non-linear dynamics help

forecasting?', In *International Journal of Forecasting, 12, 465-473.*

[35] Computational Finance, (2000), 'The 7th international conference (31 May - 2 June 2000). A 3-day Joint Conference with the Forecasting Financial Markets Conference FFM2000. A-P. N. Refenes, A Timmermann, C. Dunis, J. Moody, A. N. Burgess', *http://www.london.edu/cf2000/.*

[36] Computational Finance, (1999), 'The 6th international conference (6 - 8 January 1999). Prof. Yaser S. Abu-Mostafa, Prof. Blake LeBaron, Prof. Andrew W. Lo, Prof. Andreas S. Weigend ', *The MIT Press, Cambridge Massachusetts, London England.*

[37] Chan, L. & Lakonishok, J., (1997), 'Institutional Equity Trading Costs: NYSE versus Nasdaq', In *Journal of Finance, vol.52, No.2 (June): 713-35.*

[38] Choey, M. & Weigend, A. S., (1997), 'Nonlinear Trading Models through Sharpe Ratio Maximization', *Working Paper IS-97-005, Leonard N. Stern School of Business, New York University.*

[39] Choi, H.-G., Principe, J, & Hutchinson, A., (1993), 'A multiresolution segmentation of respiratory EMG signals', In *IEEE Biomed. Eng.'94, vol. 41-13, pp. 257-66.*

[40] Chrisman, L. D., (1996), 'Approximation of Graphical Probabilistic Models by Iterative Dynamic Discretization and its applications to Time Series Segmentation', *PhD Thesis, CMU-CS-96-166, September.*

[41] Christie, W. & Schultz, P., (1994), 'Why Do NASDAQ Market Makers Avoid Odd-Eighth Quotes?', In *Journal of Finance, vol.49, No.5 (December): 1813-40.*

[42] Clements, M. P. & Hendry, D.F., (1998), 'Forecasting Economic Time Series', *Cambridge University Press.*

[43] Conrad, J. & Kaul, G., (1998), 'An Anatomy of Trading Strategies', *The Review of Financial Studies, Vol.11, No.3, pp. 489-519.*

[44] Datastream International Limited, (1997), Monmouth House, 58-64 City Road, london EC1Y 2AL .

[45] Davis, L., (1991), 'Handbook of Genetic Algorithms', *Van Nostrand Reinhold, New York.*

[46] Deboeck, G. J, (1994), 'Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets', *Wiley Finance Edition.*

[47] Dekker, L. & Kingdon, J., (1994), 'Development Needs forDiverse Genetic Algorithms Design', In *Genetic Algorithms in Optimization, Simulation and Modelling (eds. J. Stender, E. Hillebrand and J. Kingdon) IOS Press Amsterdam, pp. 9-26.*

[48] DeLurgio, S. A., (1998), 'Forecasting Principles and Applications', *Irwin/McGraw-Hill.*

[49] Diamantaras, K. I. & Kung, S. Y., (1996), 'Principal Component Neural Networks', *WIley, New York.*

[50] Dickenson, J. P., (ed.), (1974), 'Portfolio Analysis: A Book of Readings', *Saxon House, Farnborough, Hants.*

[51] Dickey, D. A. & Fuller, W.A., (1979), 'Distributions of the estimators for autoregressive time series with unit root', In *Journal of the American Statistical Association.*

[52] Engle, R.F. & Granger, C. W. J., (1987), 'Co-integration and Error Correction: Representation, Estimation and Testing', In *Econometrica, Vol. 55, No 2, pp. 251-76.*

[53] Siatras, J., (2000), *Money and the Market, July 2000 (in Greek).*

[54] Faloutsos, C., Ranganathan, M, & Manolopoulos, Y., (1994), 'Fast Subsequence Matching in Time Series Databases', In *SIGMOD'94*.

[55] Fama, E. & French, K., (1988), 'Permanent and temporary components of stock prices', In *Journal of Political Economy, 96, 246-73.*

[56] Fancourt, C. & Principe, J., (1996), 'A neighborhood map of competing one step predictors for piecewise segmentation of the time series', In *Proc. of ICNN'96, vol. 4, pp. 1906-11.*

[57] Feldman, J., Cooper, L., Koch, C., Lippman, R., Rumelhart, D., Sabbah, D. & Waltz, D., (1990), 'Connectionist Systems', *Annual Review Computer Science, 4, 369-81.*

[58] Foster, B. F., Collopy, F. & Ungar, L., (1991), 'Neural Network Forecasting of Short Noisy Time Series', *Presented at the ORSA TIMS National Meeting, May.*

[59] Fraser, A. M. & Swinney, H. L., (1986), 'Independent Coordinates for Strange Attractors from Mutual Information', In *Physical Review A 33, pp. 1134-1140.*

[60] Fritzke, B., (1995), 'A Growing Neural Gas Network Learns Topologies', In *Advances in Neural Information Processing Systems, 7.*

[61] Fritzke, B., (1996), 'Growing Self-organizing Networks - Why?', In *ESANN'96: European Symposium on Artificial Neural Networks, Brussels, pp. 61-72.*

[62] Fritzke, B., (1997), 'A self-organizing network that can follow non-stationary distributions', In *Proceedings of ICANN97, International Conference on Artificial Neural Networks, pp.613-18, Springer Verlag.*

[63] Fritzke, B., (1998), 'Vectorbasierte Neuronale Netze (In German)', *Shaker Verlag.*

[64] Gately, E., (1996), 'Neural Networks for Financial Forecasting', *Wiley, New York.*

[65] Gencay, R., (1998), 'Optimization of Technical Trading Strategies and the Profitability in Security Markets', In *Economic Letters, Vol.59, pp.249-254.*

[66] Goldberg, D. E., (1989), 'Genetic Algorithms in Search, Optimization and Machine Learning', *Addison Wesley, Reading MA.*

[67] Goldberg, D. E. & Deb, K., (1991), 'A Comparative Analysis of Selection Schemes used in Genetic Algorithms', In *Foundations of Genetic Algorithms (eds. G. Rawlins), Morgan Kaufmann. San Mateo CA, pp. 69-93.*

[68] Goldin, D. Q., & Kanellakis, P. C., (1995), 'On Similarity Queries for Time Series Data: Constraint Specification and Implementation', In *International Conference on the Principles and Practice of Constraint Programming, pp.137-153.*

[69] Goonatilake, S. & Feldman, K., (1994), 'Genetic Rule induction for Financial Decision Making', In *Genetic Algorithms in Optimization, Simulation and Modelling (eds. J. Stender, E. Hillebrand and J. Kingdon) IOS Press Amsterdam, pp. 185-201.*

[70] Granger, C. W. J., (1999), 'Empirical Modelling in Economics: Specification and Evaluation', *Cambridge University Press.*

[71] Grassberger, P. and Procaccia, I., (1983), 'Measuring the Strangeness of Strange Attractors', *Physica D 9, pp. 189-208.*

[72] Hargreaves, C. P., (1994), 'Nonstationary time series analysis and co-integration', *Oxford University Press.*

[73] Hill, T., O'Conner, A. M., Marquez, L. & Remus, A. W., (1992), 'Neural Network Models for Forecasting: A Review', In *Proceedings of the 25th Hawaii International Conference on*

System Sciences, Vol.4.

[74] Hofmann, J, B.., (1950), 'Etymologishes Wörterbuch des Griechischen', *München, 1950.*

[75] Hottelling, H., (1933), 'Analysis of a complex of statistical variables into principal components', In *Journal of Educational Psychology, 24, pp. 417-441 and 498-520, (pp. 417-441 are also in Bryant and Atchley, 1975).*

[76] Hsieh, D. A., (1988), 'The statistical properties of daily foreign exchange rates:1974-1983', *Journal of international economics, 24, pp.129-145.*

[77] Hsieh, D. A., (1989), 'Testing for nonlinear dependance in daily foreign exchange rates', *Journal of Business, vol. 62, no. 3.*

[78] Huang, R. & Stoll, H., (1996), 'Dealer versus Auction Markets: A Paired Comparison of Execution Costs on NASDAQ and the NYSE', In *Journal of Financial Economics, vol.41, No.3 (July): 313-58.*

[79] Hurst, H. E. 'The Long Term Storage Capacity of Reservoirs', In *Transactions of the American Society of Civil Engineers, 116*, 1951.

[80] Hyvärinen, A., (1999), 'Survey on Independent Component Analysis', In *Neural Computing Surveys, Vol. 2, pp. 94-128.*

[81] Hyvärinen, A. & Oja, E., (1999), 'Independent Component Analysis: A Tutorial', *A version of this paper will appear in Neural Networks with the title "Independent Component Analysis: Algorithms and Applications".*

[82] Itakura, F., (1975), ", In *IEEE Trans. Acoust. Speech and Signal Process., ASSP-23(1), 67.*

[83] Jackson, E., (1991), 'A User's Guide to Principal Components', *John Wiley & Sons, NY.*

[84] Johansen, S, (1988), 'Statistical analysis of co-integration vectors', In *Journal of Economic Dynamics and Control, 12: 131-154.*

[85] Johnston, J., (1984), 'Econometric Methods.', *3rd edn. Singapore: McGrawHill.*

[86] Jolliffe, I.T., (1986), 'Principal Component Analysis', *Springer Verlag, New York.*

[87] Judd, K. & Mees, A. I., (1995), 'On selecting models for non-linear time series', In *Physica D, 82:426-444.*

[88] Kang, J., Choey, M., & Weigend, A. S., (1996), 'Nonlinear Trading Models and Asset Allocation Based on Sharpe Ratio Optimization.', In *Proceedings of the Neural Networks in the Capital Markets (NNCM'96).*

[89] Karhunen, K., (1946), 'Zur Spektraltheorie stochasticher Prozesse', In *Ann. Acad. Sci. Fenn., 34.*

[90] Kehagias, A. & Petridis, V., (1997), 'Time Series Segmentation using Predictive Modular Neural Networks', *Neural Computation, Vol. 9, pp. 1691-1710.*

[91] Kennel, M., Brown, R. & Abarbanel, H., (1992), 'Determing Embedding Dimension for Phase-Space Reconstruction', In *Physical Review, A 45, pp. 3403-3411.*

[92] Keogh, E., (1997), 'Fast Similarity Search in the Presence of Longitudinal Scaling in Time Series Databases', In *Proceedings of the 9th International Conference on Tool with Artificial Intelligence, pp. 578-584, IEEE Press.*

[93] Keogh, E. & Pazzani, M., (1999), 'Scaling up dynamic time warping to massive datasets', In *3rd European Conference on Principles and Practice of Knowledge Discovery in*

Databases..

[94] Kim, M. J., Nelson, C. R. & Startz, R., (1991), 'Mean reversion in stock prices? A reappraisal of empirical evidence', In *Review of Economic Studies, 58, pp. 515-528.*

[95] Kimoto, T., Asakawa, K., Yoda, M. & Takeoda, M., (1990), 'Stock Market Prediction System with Modular Neural Networks', In *Proceedings of the International Joint Conference on Neural Networks, Vol. 1, San Diego, pp. 1-6.*

[96] Kingdon, J. & Feldman, K., (1995), 'Genetic Algorithms and some applications in Finance', *Journal of Applied and Mathematical Finance,1,1.*

[97] Kingdon, J., (1997), 'Intelligent Systems and Financial Forecasting', In *Perspectives in Neural Computing Series, Springer Verlag.*

[98] Koop, G., (2000), 'Analysis of Economic Data', *John Wiley & sons.*

[99] Kruskal, J & Liberman, M., (1983), 'The symmetric time warping problem: From continuous to discrete', In *Time warps, string edits and macromolecules: The theory and practice of sequence comparison, ed.D.Sankoff and J.Kruskal.*

[100] Lades, M., Vorbrueggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Wuertz, R. P., & Konen, W., (1993), 'Distortion Invariant Object Recognition in the Dynamic Link Architecture', In *IEEE transactions on Computers, 42:300-311.*

[101] Lapedes, A. S. & Farber, R., (1987), 'Non-linear Signal Processing using Neural Networks: Prediction and System Modeling', *Technical Report LA-UR-87, Los Alamos National Laboratory.*

[102] Larsen, C. & Lam, L., (1992), 'Chaos and the Foreign Exchange Market', In *Lam, L. and Naroditsky, V. (eds.), Modelling Complex Phenomena, Springer, New York.*

[103] Lisi, F. & Medio, A., (1997), 'Is a Random Walk the best Exchange-Rate Predictor?', In *International Journal of Forecasting, 13, 255-267.*

[104] Lo, A. W. & MacKinlay, A. C., (1988), 'Stock market prices donot follow random walks: Evidence from a simple specification test', In *Review of Financial Studies, 1, pp. 41-66.*

[105] Lo, A. W. & MacKinlay, A. C., (1990), 'When are contrarian profits due to stock market overreaction?', In *Review of Financial Studies, 2, pp. 175-206.*

[106] Lo, A. W. & MacKinlay, A. C., (1999), 'A Non-Random Walk Down Wall Street', *Princeton University Press, Princeton, New Jersey.*

[107] Loeb, T., (1983), 'Trading Cost: The Critical Link between Investment Information and Results', In *Financial Analysts Journal, vol.39, No.3 (May/June): 39-44.*

[108] Lorenz, E. N., (1963), 'Deterministic non-periodic flow', In *J. Atmos Sci., 48:1752.*

[109] Maddala, G. S. & Kim, I. M., (1998), 'Unit Roots, Cointegration and Structural Change', *Cambridge University Press.*

[110] Makridakis, S., Anderson, A., Carbone, R., Fildes, R., Hilbon, M., Lewandowski, R., Newton, J., Parzen, E. & Winkler, R., (1982), 'The Accuracy of Extrapolation Methods: results of a forecasting competition', *Journal of Forecasting, 1, pp. 111-53.*

[111] Malkiel, B., (1996), 'A Random Walk Down Wall Street', *W.W. Norton & company, NY, London.*

[112] Mandelbrot, B. B. and Wallis, J. R., (1969), 'Robustness of the rescaled range R/S in the measurement of noncyclic long run statistical dependence', In *Water Resources Research, 4 967-88.*

[113] Martinetz, T. M., Berkovich, S. G. & Schulten, K. J., (1992), 'Neural-Gas Network for Vector Quantization and its Application to Time-Series Prediction', In *IEEE Transactions on Neural Networks, Vol. 4 NO. 4 July 1992.*

[114] Mees, A. I., Rapp, P. E., and Jennings, L. S., (1987), 'Singular Value Decomposition and Embedding Dimension', In *Phys. Review A36, 340-346.*

[115] Meulbroeck, L. K., (1992), 'An empirical analysis of illegal insider trading', In *Journal of Finance, 47, pp. 1661-72.*

[116] Moody, J. E., (1992), 'The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems', In *Advances in Neural Information Processing Systems, Vol. 4, (eds, J.E. Moody et al.), Morgan Kaufmann, San Mateo CA, pp. 847-54.*

[117] Moody, J. E. & Darken, C, (1989), 'Fast learning in networks of locally-tuned processing units', In *Neural Computation, 1:281-294.*

[118] Moody, J. E. & Utans, J., (1992), 'Principled Architecture Selection for Neural Networks: Application to Corporate Bond Rating', In *Advances in Neural Information Processing Systems, Vol. 4.*

[119] Moody, J. & Wu, L. Z., (1996), 'Optimization of Trading Systems and Portfolios', In *Proceedings of the Neural Networks in the Capital Markets (NNCM'96).*

[120] Moody, J., Wu, L., Liao, Y., & Saffell, M., (1998), 'Performance Functions and Reinforcement Learning for Trading Systems and Portfolios', In *Journal of Forecasting,*

vol.17, pp. 441-470.

[121] Murphy, J. J., (1999), 'Technical Analysis of the Financial Markets: A comprehensive guide to trading methods and applications', *NYIF, Paramus.*

[122] Myers, C., Rabiner, L. R. & Rosenberg, A. E., (1980), '??', In *IEEE trans, Acoust. Speech and Signal Process, ASSP-28(6),623.*

[123] Nelson, S. A., (1903), 'ABC of Stock Market Speculation', *First published in 1903, Reprinted 1978 by Frasier Publishing Co..*

[124] Nelson, D. B. & Foster, D.P., (1992), 'Assympotic filtering theory for univariate ARCH models', *Econometrica, 62, pp.1-41, In modelling stock market volatility, edited by P.E. Rossi, Academic Press, 1996.*

[125] Nelson, D. B., (1989), 'Modelling Stock Market Volatility Changes', *ASA 1989, in Proceedings of the Business and Economic Statistics Section, pp.93-98, edited by P.E. Rossi, Academic Press, 1996.*

[126] Ntungo, C., & Boyd, M., (1998), 'Commodity Futures Trading Performance using Neural Network Models versus ARIMA Models', In *Journal of Futures Markets, Vol.18, No.8, pp. 965-983.*

[127] Oppenheim, A. V. & Schafer, R. W., (1975), 'Digital Signal Processing', *Prentice Hall, Englewood Cliffs, N.J..*

[128] Ott, E., Sauer, T. and Yorke,J. A., (1994), 'Copying with Chaos', *John Wiley & Sons, Inc..*

[129] Papageorgiou, C. P., (1997), 'High Frequency Time Series Analysis and Prediction using Markov Models', In *Proceedings of CIFEr'97 Conference, New York.*

[130] Pearson, K., (1901), 'On lines and Planes of closest fit to points in space', In *Philosophical Magazine, 2 (sixth series): pp. 559-572.*

[131] Pesaran, M. H. & Timmermann, A., (1994), 'Forecasting Stock Returns. An Examination of Stock Market Trading in the Presence of Transaction Costs', In *Journal of Forecasting, 13, pp. 330-65.*

[132] Pesaran, M. H. & Timmermann, A., (1995), 'The Robustness and Economic Significance of Predictability of Stock Returns', In *Journal of Finance, 50, pp. 1201-28.*

[133] Pesaran, M. H. & Timmermann, A., (1998), 'A Recursive Modelling Approach to Predicting UK Stock Returns', .

[134] Peters, E. E., (1999), 'Patterns in the Dark: Undersatnding Risk and Financial Crisis with Complexity Theory', *John Whiley & Sons, New York.*

[135] Peters, E. E., (1994), 'Fractal Market Analysis: Applying Chaos Theory to Investment and Economics', *John Whiley & Sons, New York.*

[136] Phillips, P. C. B., (1987), 'Time series regression with a unit root'. In *Econometrica, Vol.55, No 2, pp. 277-301.*

[137] Phillips, P. C. B. & Ouliaris, S., (1988), 'Testing for cointegration using Principal Component methods', In *Journal of Economic Dynamics and Control, 12: 105-30.*

[138] Phillips, P. C. B. & Perron, P., (1988), 'Testing for a unit root in time series regression', In *Biometrica, Vol 75, pp. 335-46.*

[139] Portugal, M. S., (1994), 'Neural Networks versus Time Series Methods: A Forecasting Exercise', *presented at the 14th International Symposium of Forecasting, Stockholm School of Economics, Stockholm, Sweden.*

[140] Poterba, J. & Summers, L., (1988), 'Mean reversion in the stock prices: Evidence and implications', In *Journal of Financial Economics, 22, pp. 27-59.*

[141] Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P., (1992), 'Numerical Recipes in C: The Art of Scientific Computing', *2nd Edition, Cambridge University Press.*

[142] Pring, M. J., (1991), 'Technical Analysis Explained', *3rd edition, McGrawHill.*

[143] Pring, M. J., (1998), 'Martin Pring's Introduction to Technical Analysis', *Finance Series, McGrawHill, International Edition.*

[144] Rabiner, L. & Juang, B.-H., (1993), 'Fundamentals of speech recognition', *Prentice Hall.*

[145] Refenes, A. N., (1993), 'Constructive learning and its application to currency exchange rate forecasting', In *Trippi and Turban (eds), Neural networks in finance and investing: using artificial intelligence to improve real-world performance, revised edition 1996.*

[146] Refenes, A. N., (1994), 'Neural Networks in the Capital Markets', *John Wiley, Chichester.*

[147] Refenes, A.-P. N. & Zapranis, A. D., (1999), 'Neural Model Identification, Variable Selection and Model Adequacy', In *Journal of Forecasting, 18, pp. 299-332.*

[148] Ridley, M., (1993), 'Mathematics of Markets', *Economist Survey: Frontiers of Finance, 9th of October.*

[149] Rorabach, C. B., (1997), 'Digital Filter Designer's Handbook', *McGraw-Hill.*

[150] Said, S. E. & Dickey, D. A., (1984), 'Testing for unit roots in autoregressive moving average models of unknown orders', In *Biometrica, Vol. 71, pp. 599-607.*

[151] Sakoe, H. & Chiba, S., (1978), '??', In *IEEE Trans. Acoust. Speech and Signal Process., ASSP-26(1), 43.*

[152] Sakoe, H. and Chiba, S., (1990), 'Dynamic programming algorithm optimization for spoken word recognition', In *Alex Waibel and Kai-Fu Lee, editors, Reading in speech recognition, pages 159-165. Morgan Kaufmann Publishers, Inc.,San Mateo California.*

[153] Samuelson, P. A., Koopmans, T. C. & Stone, J. R. N., (1954), 'Report of the evaluative comitee for Econometrica', *Econometrica, 22, 141-6.*

[154] Sargan, J. D., (1964), 'Wages and prices in the United Kingdom', In *Econometric Analysis for Economic Planning, (Mort, P. E., Mill, G. and Whitaker, J. K. eds), London; Butterfield pp. 25-54.*

[155] Sargan, J. D. & Bhargava, A., (1983), 'Testing residuals from least squares regression for being generated by the Gaussian random walk.', In *Econometrica 51: 153-174.*

[156] Sastri, T., English, J. & Wang, G., (1990), 'Neural Network for Time Series Model Identification and Chage Detection', *Texas A&M University Working Paper INEN/QC/WP/03/07-90.*

[157] Sauer, T, Yorke, J. A. & Casdagli, M., (1991), 'Embedology', In *Journal of Statistical Physics, 65, pp. 579-616.*

[158] Schoelkopf, B., (1997), 'Support Vector Learning', *PhD Thesis, Technical University Berlin.*

[159] Schwager, J. D., (1999), 'Getting Started in Technical Analysis', *John Wiley & Sons.*

[160] Sharda, R. & Patel, R., (1990), 'Neural Networks as Forecasting Experts: An empirical test', In *Proceedings of the 1990 UCNN meeting, Vol.2, pp. 491-4.*

[161] Sharda, R. & Patel, R., (1990), 'Connectionist Approach on Time Series Prediction', *Oklahoma State University Working Paper, 90-26.*

[162] Shazly, M. & Shazly, H., (1997), 'Comparing the Forecasting Performance of Neural Networks in Forward Exchange Rates', In *Journal of the Multinational Financial Management, 7, pp. 345-56.*

[163] Soofi, A. S. & Cao, L., (1999), 'Nonlinear Deterministic Forecasting of Daily Peseta-Dollar Exchange Rates', In *Economics Letters, 62, 175-180.*

[164] Srikant, R. & Agrawal, R., (1995), 'Mining Sequencial Patterns: Generalizations and Performance Improvements', *Research Report RJ 9994, IBM Almaden Research Center, Jan Jose, California.*

[165] Stanley, L. S., (1983), 'Time Series Segmenation: A Model and a Method', *Information Sciences, 29, pp. 7-25.*

[166] Stanley, L. S., (1983), 'On Segmentation of Time Series', *Studies in Econometrics, Time Series and Multivariate Statistics (S. Kalvin, T. Amemiya and L. Goodman, eds.), Academic Press, pp. 311-330.*

[167] Stender, J., Hillebrand, E. & Kingdon, J.(eds), (1994), 'Genetic Algorithms in Optimization, Simulation and Modelling', *IOS Press, Amsterdam, pp. 9-29.*

[168] Takens F., (1981), 'Detecting Strange Attractors in Fluid Turbulence', In *Rand, D. A. & Young, L. S. (Eds.) Dynamical Systems and Turbulence, Lecture Notes in Mathematics,*

Vol. 898, Springer Verlag, Berlin.

[169] Tang, Z., de Almeida, C. & Fishwick, P. , (1990), 'Time Series Forecasting using Neural Networks vs. Box Jenkins methodology', *Presented at the International Workshop on Neural Networks, February.*

[170] Thomas, R. L., (1997), 'Modern Econometrics: an introduction', *Addison-Wesley.*

[171] Towers, N. & Burgess, A. N., (1998), 'Optimisation of Trading Strategies using Parameterised Decision Rules', In *Perspectives on Financial Engineering and Data Mining, PRoc. IDEA'98, edited by L. Xu et al., Springer-Verlag.*

[172] Triesch, J. & C. von der Malsburg, (1997), 'Robotic Gesture Recognition', In *GW'97, Gesture Workshop, Bielefeld, Germany, Lecture Notes in Artificial Intelligence, 1371, pp.233-244.*

[173] Trippi, R. & DeSieno, D., (1992), 'Trading Equity Index Futures with a Neural Network', In *Journal of Portfolio Management, Vol.19, pp.27-33.*

[174] Trippi, R. & Turban, E., (1996), 'Neural networks in finance and investing: using artificial intelligence to improve real-world performance', *Revised edition McGraw-Hill.*

[175] Utans, J., Holt, W. T. & Refenes, A. N., (1997), 'Principal Component Analysis for modelling Multi-currency portfolios', In *Decision Technolgies in Financial Engineering, eds. Weigend et al., World Scientific.*

[176] Wagner, W. & Edwards, M., (1993), 'Best Execution', In *Financial Analysts Journal, vol.49, No.1 (January/February): 65-71.*

[177] Weigend, A. S., Huberman, B. A., & Rumelhart, D. E., (1990), 'Predicting the future: A connectionist approach', *Stanford PDP Research Group, Technical Report, PDP-90-01.*

[178] Weigend, A. S., Huberman, B. A., & Rumelhart, D. E., (1992), 'Predicting Sunspots and Exchange Rates with Connectionist Networks', In *M. Casdagli and S. Eubank eds., Nonlinear Modelling and Forecasting, SFI studies in the Science of Complexity, Proc. vol. XII. pp. 395-432.*

[179] Weigend, A. S., Shi, S., (1998), 'Predicting Daily Probability Distributions of the S&P 500 Returns', *Working paper IS-98-23 (Information Systems) and S-98-36 (Salomon Center), Leonard N. Stern School of Business, New York University.*

[180] White, H., (1988), 'Economic Prediction using Neural Nets: The case of the IBM daily stock returns', In *Proceedings of IEEE International Conference on Neural Networks, Vol.2, IEEE, pp. 451-8.*

[181] Wiskott, L., Fellous, J.-M., Krueger, N, & c. von der Malsburg, (1995), 'Face Recognition and Gender Determination', *Int. Workshop on Automatic Face- and Gesture- Recognition, Zurich.*

[182] Wiskott, L., Fellous, J.-M., Krueger, N, & c. von der Malsburg, (1996), 'Face Recognition by Elastic Graph Matching', *IEEE - PAMI.*

[183] Wolf, A., Swift, J. B., Swinney,H. L., & Vastano, J. A., (1985), 'Determing Lyapunov Exponents from a Time Series', In *Physica D 16, pp 285-317.*

[184] Xu, L., & Cheung, Y., (1997), 'Adaptive Supervised Learning Decision Networks for Traders and Portfolios', In *Journal of Computational Intelligence in Finance, (November-December): 11-16.*

[185] Zapranis, A. & Refenes, A.-P., (1999), 'Principles of Neural Model Identification, Selection and Adequacy; With applications to financial econometrics'. In *Perspectives in Neural Computing, Springer Verlag.*