Independent Component Analysis and Feature Extraction of Financial Time Series

YANNICK CAILLÉ

Master of Science by Research
in
Pattern Analysis and Neural Networks

Supervisor: Professor David LOWE

ASTON UNIVERSITY

September 1998

Aston University

Independent Component Analysis and Feature Extraction of Financial Time Series

Yannick CAILLÉ

Master of Science by Research
in
Pattern Analysis and Neural Networks, 1998

Thesis Summary

This thesis discusses the application of a modern signal processing technique known as independent component analysis (ICA) or blind source separation to univariate time series. To perform single channel ICA on this univariate time series, we work within the embedding framework, using Takens' delay coordinate maps. After a brief presentation of the results obtained with PCA (Signal/Noise decomposition, dimensionality reduction), we show that the same kind of experiments can be done with ICA. Studies done so far have yielded encouraging results among which the following emerge as the most noteworthy :

- ICA, just like PCA, preserves the possibility to perform a Signal/Noise decomposition.
- Independent components (ICs) reveal evidence of clustering amongst them.
- The possibility to efficiently rank the ICs.

Using all these results, we show that the time series can be reconstructed surprisingly well by using a small number of weighted ICs. Independent component analysis seems to be a promising powerful method of analyzing and understanding driving mechanisms in financial markets.

Keywords: Feature extraction, Principal Component Analysis, Independent Component Analysis, Dynamical Embedding, Delay Coordinate Maps, Clustering, Financial Time Series

Acknowledgements

First, I would like to thank my supervisor, Professor David Lowe, for the wonderful support he has given me over the year. He has been the biggest source of encouragement for this project, and his comments and suggestions were absolutely indispensible. I especially thank him for making time for me when he had very little of his own. He also deserves credit for originally getting me into a new field, that is, ICA applied to financial time series.

I would also like to give special acknowledgement to Nathalie Noël and Ernest Fokoué for constantly providing me with their friendly advice and support, and for creating a stimulating and pleasant atmosphere throughout the academic year.

I am especially indebted to my parents whose unremitting efforts and sacrifices made it possible for me to come and study in England.

Finally, I am grateful to Jean-François Cardoso for making the source code of the `JADE` algorithm available.

Contents

1 Introduction **8**
- 1.1 Independent Component Analysis in finance . . . 8
- 1.2 Time Series Analysis . . . 9
- 1.3 Thesis Structure . . . 10

2 Basic concepts **13**
- 2.1 Principal Component Analysis . . . 13
- 2.2 The Concept of Independence . . . 14
- 2.3 Linear Model of Independent Component Analysis . . . 16
- 2.4 Removing Correlations . . . 17
- 2.5 Cumulants and Cumulant Matrices . . . 18

3 Dynamical Embedding **22**
- 3.1 Dynamical Systems . . . 22
- 3.2 The Embedding Theorem . . . 23
- 3.3 Delay Coordinate Maps . . . 24
- 3.4 Degrees of Freedom . . . 26
- 3.5 Experimental Results . . . 28
  - 3.5.1 General comments about the data . . . 28
  - 3.5.2 Determining the embedding delay τ . . . 28
  - 3.5.3 Determining the window size of the embedding . . . 28
  - 3.5.4 Determining the degrees of freedom . . . 30
  - 3.5.5 Shape of the attractor . . . 30

4 Independent Component Analysis **36**
- 4.1 General Overview of ICA . . . 36
- 4.2 Entropy Maximization . . . 38
- 4.3 Fixed-Point Algorithms . . . 38
- 4.4 The Bigradient Algorithm . . . 40
- 4.5 Joint Approximate Diagonalization of Eigenmatrices . . . 41
  - 4.5.1 The algorithm . . . 41
  - 4.5.2 A toy problem . . . 43

5 ICA Results **45**
- 5.1 General Information Concerning ICA . . . 45
  - 5.1.1 Reasons to use ICA in finance . . . 45
  - 5.1.2 Description of the data . . . 46

5.1.3 Structure of the independent components 46
5.2 Reconstruction of the Time Series 47
5.2.1 Reconstruction algorithm 47
5.2.2 Sorting the independent components 48
5.2.3 Interpretation of the results 50
5.2.4 Reconstruction using clusters 53
5.3 Thresholded Reconstruction 57
5.4 Comparison with PCA 59

6 Conclusions **60**
6.1 Dynamical Embedding 60
6.2 ICA 60
6.3 Limitations of the ICA model 61
6.4 Conclusion 61
6.5 Future Work 62

References **63**

A A Joint Diagonalization Algorithm **66**

B Reconstruction Using Raw Data **68**

List of Figures

2.1 Basic representation of ICA 16

3.1 An illustration of Takens' theorem 25
3.2 British pound futures contract data 29
3.3 Eigenspectra for different window sizes 31
3.4 Eigenspectrum of the delay vectors 32
3.5 Initial Time Series 33
3.6 Projection into 3D PCA space 34

4.1 Toy problem for `JADE` 44

5.1 `JADE` (20 first independent components) 47
5.2 Reconstruction of the time series 49
5.3 Illustration of clustering 50
5.4 Sorting criterion : $\|\tilde{\mathbf{Z}}_i - \mathbf{Z}\|$ 51
5.5 Reconstruction error 52
5.6 Reconstruction with 6 ICs 53
5.7 Reconstruction error (6 ICs) 54
5.8 Reconstruction with 5 ICs 55
5.9 Reconstruction error (5 ICs) 56
5.10 Results after Thresholding 58
5.11 Reconstruction with PCs 59

B.1 `JADE` (20 first independent components) 69
B.2 `JADE` (20 first independent vectors) 70
B.3 Reconstructed time series 71

Abbreviations and notations

ICA	Independent component analysis
PCA	Principal component analysis
$\underline{\mathbf{0}}$	A vector of zeros
$\underline{\mathbf{1}}$	A vector of ones
$\alpha, \beta, \gamma, \ldots$	Scalars
λ, λ_i	Lagrange multipliers
ξ, ξ_i	Eigenvalues
$\underline{\mathbf{e}}, \underline{\mathbf{e}}_i$	Eigenvectors
$\underline{\mathbf{a}}, \ldots, \underline{\mathbf{e}}$	Vectors
a_i	i^{th} component of vector $\underline{\mathbf{a}}$ or i^{th} column vector of matrix A
$\underline{\mathbf{s}}$	Source vector
s_i	One source
$\underline{\mathbf{v}}$	Whitened data vector
x_i, y_i	Random variables or i^{th} component of random vectors
$A, B, C, \ldots$	Matrices or constants
A	ICA mixing matrix
D	A diagonal matrix of eigenvalues
P	A matrix of eigenvectors
$\tilde{P}$	An orthogonal matrix
$Q_{\underline{\mathbf{x}}}(M)$	Cumulant matrix for random vector $\underline{\mathbf{x}}$ and matrix M
V	A whitening matrix
W	ICA separating matrix
$\mathbf{Z}$	An embedding matrix
$\mathcal{L}(\mathbb{R}^m, \mathbb{R}^n)$	Linear transformation from $\mathbb{R}^m$ to $\mathbb{R}^n$ ($n \times m$ matrices)
$i, \ldots, n$	Natural numbers
$\underline{\mathbf{a}}(k)$	k^{th} iteration of $\underline{\mathbf{a}}$
n	The number of sources
f, g, h	Functions
f_x	Density function of random variable x
f_{xy}	The joint density function of random variables x and y
$\mathbf{E}[.]$	Expected value
$\mathbf{var}\{.\}$	Variance
$\mathbf{cov}\{.\}$	Covariance
$\mathbf{cum}(.)$	Cumulant
$M^{\#}$	Pseudo-inverse of matrix M

Chapter 1

Introduction

The search for underlying deterministic forces which guide financial markets, and the construction of models which forecast these forces has provided the focus of many research papers. Traditionally these tasks have been approached by exploiting a variety of statistical tools. However, financial markets have three characteristics – noise, nonstationarity and nonlinearity. This suggests the use of unsupervised learning techniques in order to capture the nonlinearity inherent to the time series.

1.1 Independent Component Analysis in finance

It is also well known that financial time series often originate from different and independent sources, such as foreign politics or microeconomic variables or traders decisions.

Under the assumption that financial time series are generated according to an underlying (and unknown) nonlinear process, there is a need to examine techniques which might be capable of extracting the "nonlinear" principal components, or the statistically independent components of a process.

In this thesis we want to apply and compare different extensions to the Principal Component Analysis, for instance ICA, to financial time series within the embedding framework. The aim thereby is primarily to seek dimensionality reduction of the data in order to extract characteristic features from the data. Projections of the data on to these bases allow a reparameterisation of the dynamics which then might be subsequently more amenable to traditional modelling techniques. A side effect of this process should be to allow a noise reduction of the data which should also have the effect of simplifying subsequent modelling.

The basic process begins with an embedding into a feature space. This embedding allows us to study the dynamical properties of the underlying system, without knowing any information about the original manifold. The feature space is characterised by "principal directions". (The significance of the directions needs to be determined, as does the number). The time series data is projected into this feature space where either a clustering or a direct neural network modelling can take place. A problem with this approach is that financial time series are nonstationary. Hence the number of components and their directions may/will change over time. The relevance of Independent Components needs to be discussed in the context of financial data. Foreign exchange data will be used as input data.

1.2 Time Series Analysis

A principal feature of historical/retrospective analysis is that one of the measurement axes is time. Intrinsically, observations on a phenomenon (or process) over time are correlated with, or dependent on, their past. Thus, there are two major consequences : first, the order of observations is important and second, the assumption that consecutive observations constitute independent samples is invalid.

A set of statistical tools and techniques, referred to as *time series analysis* (TSA), has been developed to analyze data collected sequentially over time. There are actually three main goals of time series analysis. The first goal is to predict the future behaviour of a time series given information concerning its present state [*prediction*]. The second goal is to model the dynamics of a time series as a function either of its own past history (univariate TSA) and/or the history of other explanatory time series (multivariate TSA) [*modelling*]. Finally, the third goal of time series analysis is to diagnose the nature of time-related behaviour within and between time-series [*characterization*].

The emphasis of this thesis is laid on the last point – characterization – and concerns only *univariate* time series. More precisely, we will try to estimate to what extent noise is present in a time series, as well as to reconstruct a time series from a reduced number of features.

1.3 Thesis Structure

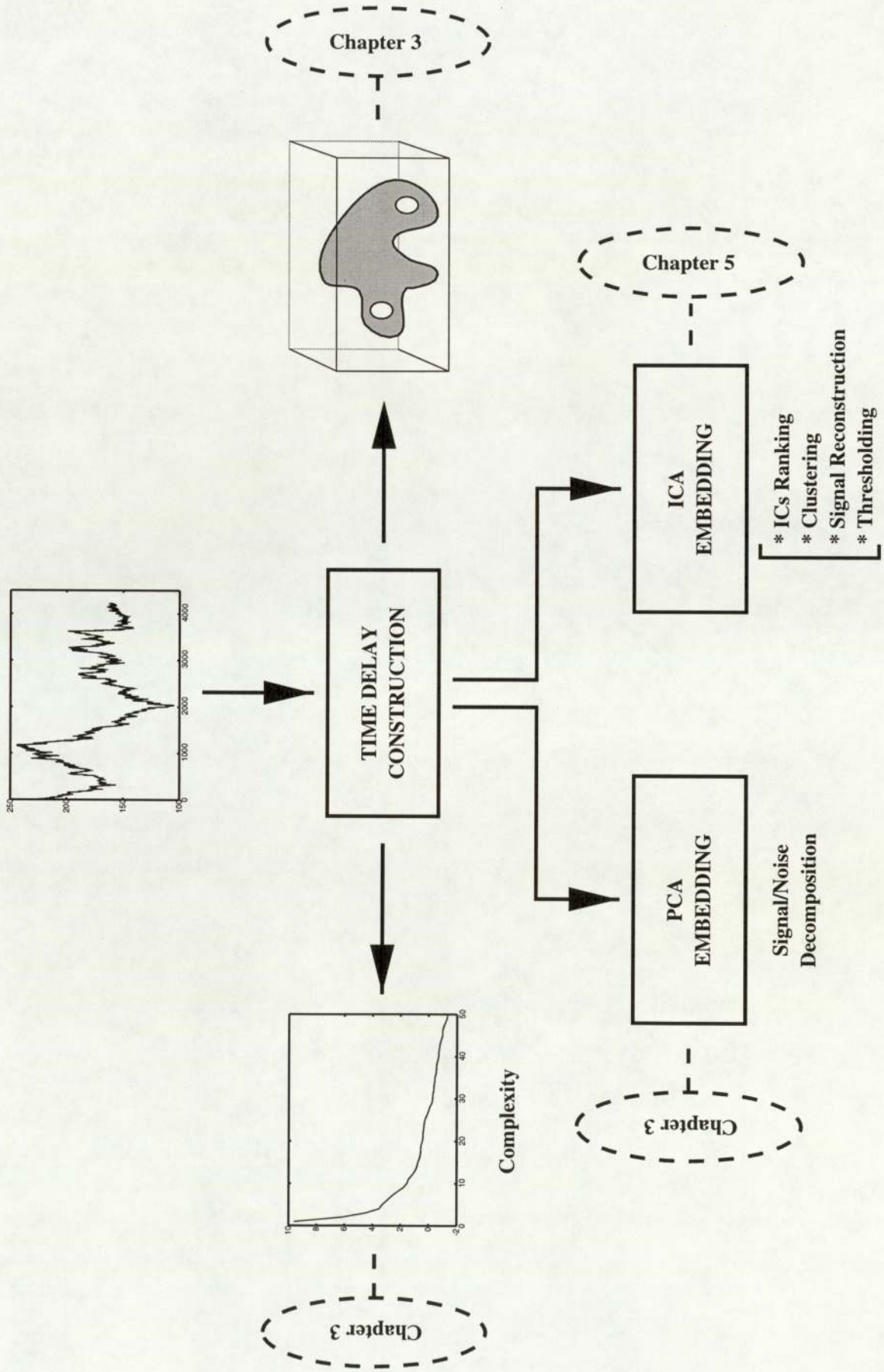
Chapter 1 gives a brief presentation of financial markets, and the inherent problems encountered when trying to study financial time series.

After explaining a few mathematical concepts required to understand the rest of this thesis in Chapter 2, we use embedology to preprocess our univariate time series. This allows us to find a few characteristics of this time series (such as the number of degrees of freedom), as well as to perform a Signal/Noise separation on this time series (using PCA).

Chapter 4 provides theoretical background to understand ICA principles, shows different ICA algorithms and illustrates the main ideas with a toy problem. The `JADE` algorithm [8] is also introduced and examined.

We use Chapter 5 to elaborate on experimental results, with particular emphasis on ICA's ability to reveal clustering, efficiently rank the ICs and perform Signal/Noise decomposition.

Finally, Chapter 6 highlights the main strengths of the methods studied, and announces ideas for future work.


Chapter 3
Chapter 5
ICA
EMBEDDING
* ICs Ranking
* Clustering
* Signal Reconstruction
* Thresholding
TIME DELAY
CONSTRUCTION
PCA
EMBEDDING
Signal/Noise
Decomposition
Complexity
Chapter 3
Chapter 3

Chapter 2

Basic concepts

2.1 Principal Component Analysis

Principal component analysis (PCA) [17] is a classical statistical method of data analysis for reducing the dimensionality of data. The purpose is to find a set of n orthogonal vectors in data space that account for as much as possible of the data variance. In terms of linear algebra, this problem consists of finding a new basis for the data so that if we drop out the least important components in the new basis, the reconstruction error is as small as possible.

To achieve this goal, we proceed as follows : the first principal component is taken to be along the direction with maximum variance. The second principal component is constrained to lie in the subspace perpendicular to the first, within which it is taken along the direction with maximum variance. Then the third principal component is taken in the maximum variance direction in the subspace perpendicular to the first two, and so on. In general, the k^{th} principal component direction is along an eigenvector direction belonging to the k^{th} largest eigenvalue of the full covariance matrix.

Let $\{(\underline{\mathbf{e}}_1, \xi_1), \ldots, (\underline{\mathbf{e}}_n, \xi_n)\}$ denote the set of eigenvector/eigenvalue pairs of $\mathbf{cov}\,\{\underline{\mathbf{x}}\}$. Let $P = [\underline{\mathbf{e}}_1 \ \ldots \ \underline{\mathbf{e}}_n]$ and $D = \mathrm{diag}(\xi_1, \ldots, \xi_n)$. If we consider the projection of the data

on the new basis, the new components are uncorrelated. As a matter of fact, the expression of the components in the new basis is given by $\underline{\mathbf{x}}_{new} = P^T \underline{\mathbf{x}}$, so

$$\begin{aligned}
\mathbf{cov}\left\{\underline{\mathbf{x}}_{new}\right\} &= \mathbf{E}\left[\underline{\mathbf{x}}_{new}\,\underline{\mathbf{x}}_{new}^T\right] \\
&= \mathbf{E}\left[P^T \underline{\mathbf{x}}\,\underline{\mathbf{x}}^T P\right] \\
&= P^T \underbrace{\mathbf{cov}\left\{\underline{\mathbf{x}}\right\}}_{=PDP^T} P \\
&= \underbrace{P^T P}_{=I} D \underbrace{P^T P}_{=I} \quad \text{(as } P \text{ is an orthogonal matrix)} \\
&= D
\end{aligned}$$

PCA is widely used in statistical signal processing, and most often, to separate the data into a pair *Signal space/Noise space.* Indeed, small eigenvalues can be associated with noise, whereas the most important ones give the dimension of the data. Thus, noise may be removed by reducing the dimension of the data.

2.2 The Concept of Independence

Given two events $\mathcal{A}$ and $\mathcal{B}$ in an event space of an experiment, we can say that $\mathcal{A}$ and $\mathcal{B}$ are *independent* if

$$\mathrm{P}(\mathcal{A}\mathcal{B}) = \mathrm{P}(\mathcal{A})\mathrm{P}(\mathcal{B})$$

Using the conditional probability $\mathrm{P}(\mathcal{B}|\mathcal{A})$ given by

$$\mathrm{P}(\mathcal{B}|\mathcal{A}) = \frac{\mathrm{P}(\mathcal{A}\mathcal{B})}{\mathrm{P}(\mathcal{A})}$$

we can see that independence implies $\mathrm{P}(\mathcal{B}|\mathcal{A}) = \mathrm{P}(\mathcal{B})$, if $\mathrm{P}(\mathcal{A}) \neq 0$. As the implication holds in the other direction, this condition is equivalent to independence. Assuming that $\mathrm{P}(\mathcal{B}) \neq 0$, we can also derived the following equivalence

$$\mathrm{P}(\mathcal{A}|\mathcal{B}) = \mathrm{P}(\mathcal{A}) \iff \mathrm{P}(\mathcal{A}\mathcal{B}) = \mathrm{P}(\mathcal{A})\mathrm{P}(\mathcal{B})$$

Actually, it is quite intuitive to think about equations $\mathrm{P}(\mathcal{B}|\mathcal{A}) = \mathrm{P}(\mathcal{B})$ and $\mathrm{P}(\mathcal{A}|\mathcal{B}) = \mathrm{P}(\mathcal{A})$ in terms of independence : if one event happens, it does not give any *additional*

information about the probability of the other event. In terms of information, independence of two events could be interpreted so that information about one event gives no additional information about the other one.

We can give a very similar interpretation of independence in the case of *random variables*. The definition of independence of two random variables x and y is

$$\mathrm{P}(x \in A, y \in B) = \mathrm{P}(x \in A)\mathrm{P}(y \in B) \quad A, B \subset \mathbb{R}$$

This is equivalent to

$$f_{xy}(x, y) = f_x(x) f_y(y)$$

if the densities exist. Thus the *conditional density*

$$f_y(y|x) = \frac{f_{xy}(x, y)}{f_x(x)}$$

becomes $f_y(y|x) = f_y(y)$. Here, information about the *value* of one random variable gives no information about the value of the other one.

The main interest in independence lies in the fact that the independent components can often be processed separately. That is, in the case of a system trying to learn some aspects of the incoming data, the separation of the data into independent components[1] allows to infer properties by examining individually the characteristics of the components and combining them in the end to form some view of the data.

Thus appears the fundamental interest in ICA for data analysis: if one could apply ICA to a complex data set, the resulting independent components could describe some underlying factors of the process. But this approach suffers from several drawbacks, particularly because the ICA linear model is too rigid, and independence is a very strong condition.

[1] i.e. components that do not interfere with each other

2.3 Linear Model of Independent Component Analysis

Let us *assume* that we have some phenomenon which manifests itself through a set of n independent random variables, and let $\underline{\mathbf{s}}$ denote the random vector which is the combination of these variables. Thus $\underline{\mathbf{s}} = [s_1 \; s_2 \ldots s_n]^T$ where $s_1, s_2, \ldots, s_n$ are called *sources* and $\underline{\mathbf{s}}$ is called the *source vector.*

Now suppose suppose that the original independent source components are observed via a *linear process.* Let $\underline{\mathbf{x}}$ be the observed random vector. Since the process is assumed linear, the relation between $\underline{\mathbf{x}}$ and $\underline{\mathbf{s}}$ can be expressed as

$$\underline{\mathbf{x}} = A\underline{\mathbf{s}} \tag{2.1}$$

The problem is to find a *demixing matrix*[2] W so that

$$\underline{\mathbf{y}} = W\underline{\mathbf{x}} \tag{2.2}$$

$$= WA\underline{\mathbf{s}} \tag{2.3}$$

where $\underline{\mathbf{y}}$ denotes a (new) set of statistically independent vectors, that are estimates of the original source signals. Figure 2.1 shows the most basic form of ICA.

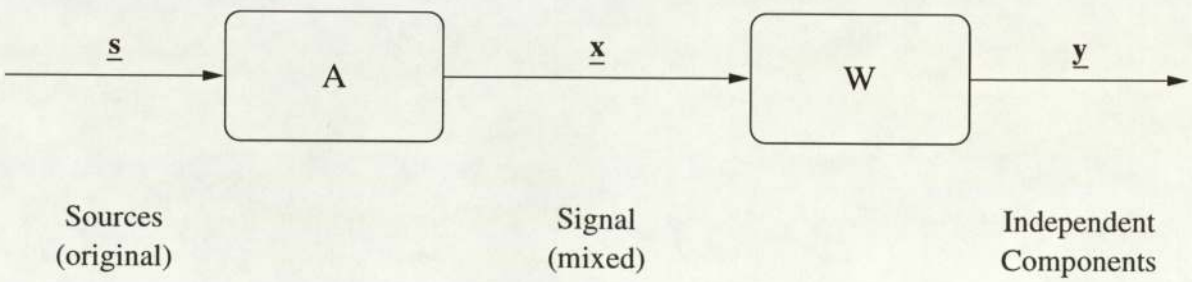


Figure 2.1: Basic representation of ICA

In the most general case, matrix $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ is refered to as a *mixing matrix,* since it mixes the independent sources. Its companion is matrix $W \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^n)$. Even though all cases $m < n, m = n$ and $m > n$ are possible, and differ significantly from each other, we assume throughout this thesis that there are as many observed signals

[2]also called *separating matrix*

as there as sources[3], that is, A is a square $n \times n$ matrix.

If $W = A^{-1}$, then $\underline{\mathbf{y}} = \underline{\mathbf{s}}$, and perfect separation occurs. In general, it is only possible to find W such that $WA = PD$ where P is a permutation matrix and D is a diagonal scaling matrix.

To find such a demixing matrix W, three main assumptions are made :

- The sources s_j are statistically independent
- There is *at most* one source with a Gaussian distribution
- The signals are stationary

2.4 Removing Correlations

Let us consider that we are dealing with zero-mean data, i.e. $\mathbf{E}[\underline{\mathbf{x}}] = \underline{\mathbf{0}}$. According to the linear model discussed in Section 2.3 (equation 2.2), the independent components of $\underline{\mathbf{s}}$ are also zero-meaned. So, let assume that our data has been *centered*[4] for the following discussion.

Given two components s_i and s_j of vector $\underline{\mathbf{s}}$, $(i, j) \in [\![1, n]\!]^2$, their covariance is

$$\begin{aligned} \mathbf{E}[s_i s_j] &= \mathbf{var}\{s_i\} = 1, \text{ if } i = j \\ \mathbf{E}[s_i s_j] &= \mathbf{E}[s_i]\,\mathbf{E}[s_j] = 0, \text{ if } i \neq j \end{aligned}$$

So, the covariance matrix of $\underline{\mathbf{s}}$ is the identity matrix, and components of $\underline{\mathbf{s}}$ are uncorrelated. Thus, we have a necessary – but not a sufficient – condition between independence and uncorrelatedness in the case of zero mean data

$$\boxed{\text{independence} \Longrightarrow \text{uncorrelatedness}}$$

[3] We shall see later that there can be repetition of sources

[4] i.e. we have zero mean data

So, it seems quite logical to think of decorrelation as a first step towards independence. We can accomplish it by transforming $\underline{\mathbf{x}}$ so that its covariance matrix will be diagonal. Furthermore, if all components have unit variance, a random vector is referred to as being *white*: this process is called *whitening* or *sphering*.

To perform whitening, we can use PCA basis vectors and variances along them. Let P denote the matrix formed of the eigenvectors of $\mathbf{cov}\{\underline{\mathbf{x}}\}$, and $D = \mathrm{diag}(\xi_1, \ldots, \xi_n)$ a diagonal matrix of corresponding eigenvalues. Let $V = D^{-1/2}P^T$ denote a *whitening matrix*, the expression of the new whitened data vector is given by

$$\underline{\mathbf{v}} = V\underline{\mathbf{x}} \tag{2.4}$$

$$= D^{-1/2}P^T\underline{\mathbf{x}} \tag{2.5}$$

We can check that $\underline{\mathbf{v}}$ is really white by computing its covariance matrix

$$\begin{aligned}
\mathbf{cov}\{\underline{\mathbf{v}}\} &= \mathbf{E}\left[D^{-1/2}P^T\underline{\mathbf{x}}\,\underline{\mathbf{x}}^T P D^{-1/2}\right] \\
&= D^{-1/2}P^T\mathbf{cov}\{\underline{\mathbf{x}}\}\,PD^{-1/2} \\
&= D^{-1/2}P^T(PDP^T)PD^{-1/2} \quad \text{(by definition of } \mathbf{cov}\{\underline{\mathbf{x}}\}) \\
&= I
\end{aligned}$$

One can notice that PCA whitening is not the only possible method for whitening, but it gives optimal (in the mean square sense) dimensionality reduction, which can be combined with the whitening operation. Nevertheless, if we consider any orthogonal matrix $\tilde{P}$ and a whitening matrix V, then $\tilde{P}V$ is also a whitening matrix, as

$$\mathbf{E}\left[(\tilde{P}V\underline{\mathbf{x}})(\tilde{P}V\underline{\mathbf{x}})^T\right] = \tilde{P}\,\mathbf{E}\left[V\underline{\mathbf{x}}\,\underline{\mathbf{x}}^T V^T\right]\tilde{P}^T = \tilde{P}\,\mathbf{E}\left[(V\underline{\mathbf{x}})(V\underline{\mathbf{x}})^T\right]\tilde{P}^T = \tilde{P}I\tilde{P}^T = I$$

2.5 Cumulants and Cumulant Matrices

Cumulants are higher-order statistics that have become increasingly popular in various signal processing tasks [26, 27]. When correlation and power spectra[5] are commonly

[5]Fourier transform of autocorrelation

used in signal processing, as 2^{nd} order statistical tools, cumulants and polyspectra[6] can be seen as the corresponding statistical tools of order higher than two.

It is particularly difficult to illustrate the properties of cumulants, as well as their main advantages. We will give briefly a definition of cumulants, and some of their properties. Let $\mathbf{cum}(x_1, \ldots, x_n)$ denote a cumulant of random variables $x_1, \ldots, x_n$. Cumulants are then defined as the coefficients of the Taylor series of the characteristic function of $\underline{\mathbf{x}} = [x_1 \ \ldots \ x_n]^T$ at point $\underline{\mathbf{0}}$, that is

$$\boxed{\mathbf{cum}(x_1, \ldots, x_n) = \frac{d}{d\omega_1} \ldots \frac{d}{d\omega_n} \Psi(\omega_1, \ldots, \omega_n) \Big|_{\omega_1=0,\ldots,\omega_n=0}}$$

where

$$\boxed{\Psi(\omega_1, \ldots, \omega_n) = \ln \mathbf{E}\left[e^{\sum_i \omega_i x_i}\right]}$$

The class of cumulants involving just one random variable is denoted by the letter κ, so that

$$\boxed{\kappa_i(x) = \mathbf{cum}(\underbrace{x, \ldots, x}_{i \text{ times}})}$$

Among the useful properties of cumulants, we can notice :

1. Cumulants are *multilinear*[7]. Let x and y denote two random variables and α a scalar

$$\begin{aligned} \mathbf{cum}(x_1, \ldots, x_i + y, \ldots, x_n) &= \mathbf{cum}(x_1, \ldots, x_i, \ldots, x_n) \\ &\quad + \mathbf{cum}(x_1, \ldots, y, \ldots, x_n) \\ \mathbf{cum}(x_1, \ldots, \alpha x_i, \ldots, x_n) &= \alpha \ \mathbf{cum}(x_1, \ldots, x_i, \ldots, x_n) \end{aligned}$$

2. If a subset of random variables $x_1, \ldots, x_n$ is independent of the others, then $\mathbf{cum}(x_1, \ldots, x_n) = 0$.

3. If $x_1, \ldots, x_n$ are Gaussian random variables and $n \geq 3$, then $\mathbf{cum}(x_1, \ldots, x_n) = 0$

[6] Fourier transforms of cumulants

[7] *multilinear* means linear for each entry

While properties 1 and 2 enable the use of cumulants as operators, property 3 makes cumulants insensitive to Gaussian noise, and thus, gives us a measure of nonnormality.

One concept which has been influential in ICA is that of 4^{th} order *cumulant matrices* [8]. A cumulant matrix $Q_{\underline{\mathbf{x}}}(M)$ is defined for a real n-dimensional random vector $\underline{\mathbf{x}}$ and an $n \times n$ matrix M as

$$\boxed{Q_{\underline{\mathbf{x}}}(M)_{ij} = \sum_{k=1}^{n}\sum_{l=1}^{n} \mathbf{cum}(x_i, x_j, x_k, x_l) m_{lk}, \;\; i,j \in \{1,\ldots,n\}} \tag{2.6}$$

Since we are working under the linear model assumption of Section 2.3, premultiplying equation (2.1) with the PCA whitening matrix V of Section 2.4 gives

$$v = Vx = VAs = Bs,$$

where v is called the *whitened data vector*, and $B \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^n)$ is the *whitened mixing matrix*. As v and s are white,

$$I = \mathbf{E}\left[vv^T\right] = \mathbf{E}\left[Bss^TB^T\right] = B\mathbf{E}\left[ss^T\right]B^T = BB^T$$

so B is orthogonal. As a consequence, matrix A can be factored as

$$A = V^{-1}B = V^{-1}\left[\underline{\mathbf{b}}_1, \ldots, \underline{\mathbf{b}}_n\right]$$

where B is unitary. Using the cumulant properties – Gaussian rejection, additivity, multilinearity – it is straightforward to establish that

$$Q_{\underline{\mathbf{v}}}(M) = \sum_{p=1}^{n} \kappa_4(s_p)\underline{\mathbf{b}}_p^T M \underline{\mathbf{b}}_p \quad \forall M.$$

This can be equivalently rewritten as

$$Q_{\underline{\mathbf{v}}}(M) = B\Gamma_M B^T \qquad \Gamma_M = \mathrm{diag}(\kappa_4(s_1)\underline{\mathbf{b}}_1^T M \underline{\mathbf{b}}_1, \ldots, \kappa_4(s_n)\underline{\mathbf{b}}_n^T M \underline{\mathbf{b}}_n).$$

This means that columns of matrix B are eigenvectors of matrix $Q_{\underline{\mathbf{v}}}(M)$ *for any matrix* M. Let β_i denote the expression $\kappa_4(s_i)b_i^T M b_i$, $i = 1, \ldots, n$. Thus, as long as

$\beta_i \neq \beta_j \; \forall \; i \neq j$ – that is, $Q_{\underline{\mathbf{v}}}(M)$ does not have identical eigenvalues – columns of matrix B can be determined up to a scalar multiplier by solving the eigenvalue problem. Then the eigenspaces are one dimensional, and the indeterminacy of the eigendecomposition precisely corresponds to the fundamental indeterminacy of ICA[8].

Even though computationally simple, this approach suffers two major drawbacks :

- We do not have any prior knowledge as how to choose M before evaluating $Q_{\underline{\mathbf{v}}}(M)$
- We use only a fraction of the 4^{th} order information

The poor statistical performance of this method (See [8]) will let us see and use a variant that works by diagonalizing many cumulant matrices simultaneously, thus trying to improve the performance (See the `JADE` algorithm in 4.5).

With the basic concepts clearly presented, we shall use the subsequent chapters to gain more insights into single channel times series analysis, with an emphasis on the exploration and study of PCA and ICA within the embedding framework.

[8]The sign/magnitude indeterminacy

Chapter 3

Dynamical Embedding

3.1 Dynamical Systems

The notion of dynamics first appeared in the fifteenth century, when Isaac Newton invented differential equations. But the initial enthusiasm concerning this invention, quickly lead to hopelessness in the way to solve them, that is, to obtain an explicit form for the solution. It is only centuries after that these equations were thought about in term of systems' behavior, by Henri Poincaré. He particularly concentrated his studies on overall behavior and stability. But it is only in the fifties that scientists were armed to discover chaotic behavior, strange attractors and fractals, with the development of high-speed computers.

One can distinguish two kinds of dynamical systems :

- Differential equations (for continuous time problems)
- Iterative maps (for discrete time problems)

A very general form for ordinary differential equations is the system $\frac{\partial \underline{\mathbf{x}}}{\partial t} = f(\underline{\mathbf{x}})$, where

$$\underline{\mathbf{x}} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$$

is the set of the system's state variables and

$$f(\underline{\mathbf{x}}) = \begin{bmatrix} f_1(x_1, \ldots, x_m) \\ \vdots \\ f_m(x_1, \ldots, x_m) \end{bmatrix}$$

describes the system's behavior. An analogous form for a general iterative map would be $x_{i+1} = f(x_i)$. The main particularity of nonlinear systems is that, for most of them, it is impossible to solve them analytically.

3.2 The Embedding Theorem

Given two spaces $\mathbf{A}$ and $\mathbf{B}$, a *mapping* between them is a function f that associates every element $\alpha \in \mathbf{A}$ with the uniquely determined element $\beta \in \mathbf{B}$. Thus, β is called the *image* of α, and α is the *preimage* of β. When spaces $\mathbf{A}$ and $\mathbf{B}$ are metric, the notions of continuity and smoothness can be introduced in that scheme. We can define then a *diffeomorphism* as a C^k mapping that is bijective. We will also define a smooth mapping f that is only injective as an *immersion*. As we are interested in preserving topological properties, we will also want the mapping to be *proper*, that is, the pre-image of every compact set is a compact set. Finally, we will define an *embedding* as a proper immersion. To simplify this idea, we can think of an embedding as being a smooth local change of coordinates : even when disfiguring a subset $\mathbf{S} \subset \mathbf{A}$, its local properties and fine structure are kept intact.

The fundamental result concerning embeddings was stated and proved by Takens, in [34], and is presented below in its original form.

Theorem 1 *Let* $\mathbf{M}$ *be a compact manifold of dimension* m.
For pairs (φ, y), $\varphi : \mathbf{M} \mapsto \mathbf{M}$ *a smooth*[1] *diffeomorphism and* $y : \mathbf{M} \mapsto \mathbb{R}$ *a smooth function, it is a generic property that the map* $\Phi_{(\varphi,y)} : \mathbf{M} \mapsto \mathbb{R}^{2m+1}$ *defined by*

$$\Phi_{(\varphi,y)}(\underline{\mathbf{x}}) = (y(\underline{\mathbf{x}}), y(\varphi(\underline{\mathbf{x}})), \ldots, y(\varphi^{2m}(\underline{\mathbf{x}})))$$

is an embedding.

For a dynamical system $\frac{\partial \underline{\mathbf{x}}}{\partial t} = f(\underline{\mathbf{x}})$, the manifold $\mathbf{M}$ would be the set of a system's state $\underline{\mathbf{x}}(t)$ for which it is very rare to find an analytic form. Thus, the map Φ is a useful way to refer to an arbitrary switch of the space's nature. Indeed, according to Theorem 1, the new set Φ carries the same fine structure as the original manifold $\mathbf{M}$. As a direct consequence to this, this new set in $\mathbb{R}^{2m+1}$ can be interpreted as the solution of a dynamical system which exhibits a behavior that is very similar to the one of the original system $\frac{\partial \underline{\mathbf{x}}}{\partial t} = f(\underline{\mathbf{x}})$. It is important to notice that the dimension of the manifold is unknown ($m = \dim(\mathbf{M})$). Nevertheless, this is not a problem as long as the embedding space – $\mathbb{R}^n$ – has embedding dimension $n \geq 2m+1$. The framework discussed here is illustrated in Figure 3.1.

3.3 Delay Coordinate Maps

Let us pick the manifold $\mathbf{M}$ to be the set of our system's state $\underline{\mathbf{x}}(t)$, where $\underline{\mathbf{x}}(t)$ represents the system's state at time t. Let $x(t) = y(\underline{\mathbf{x}}(t))$ denote our scalar observation at time t. If we chose the map $\varphi : \mathbf{M} \mapsto \mathbf{M}$ to be such that $\varphi(\underline{\mathbf{x}}(t)) = \underline{\mathbf{x}}(t+\tau)$, it will represent the system's internal dynamics. Finally, we define

$$\begin{aligned} \Phi_{(\varphi,y)} : \mathbf{M} &\longmapsto \mathbb{R}^{2m+1} \\ \underline{\mathbf{x}} &\longmapsto \big(y(\underline{\mathbf{x}}), y(\varphi(\underline{\mathbf{x}})), \ldots, y(\varphi^{2m}(\underline{\mathbf{x}}))\big) \end{aligned}$$

[1] By smooth, we mean at least $\mathcal{C}^2$

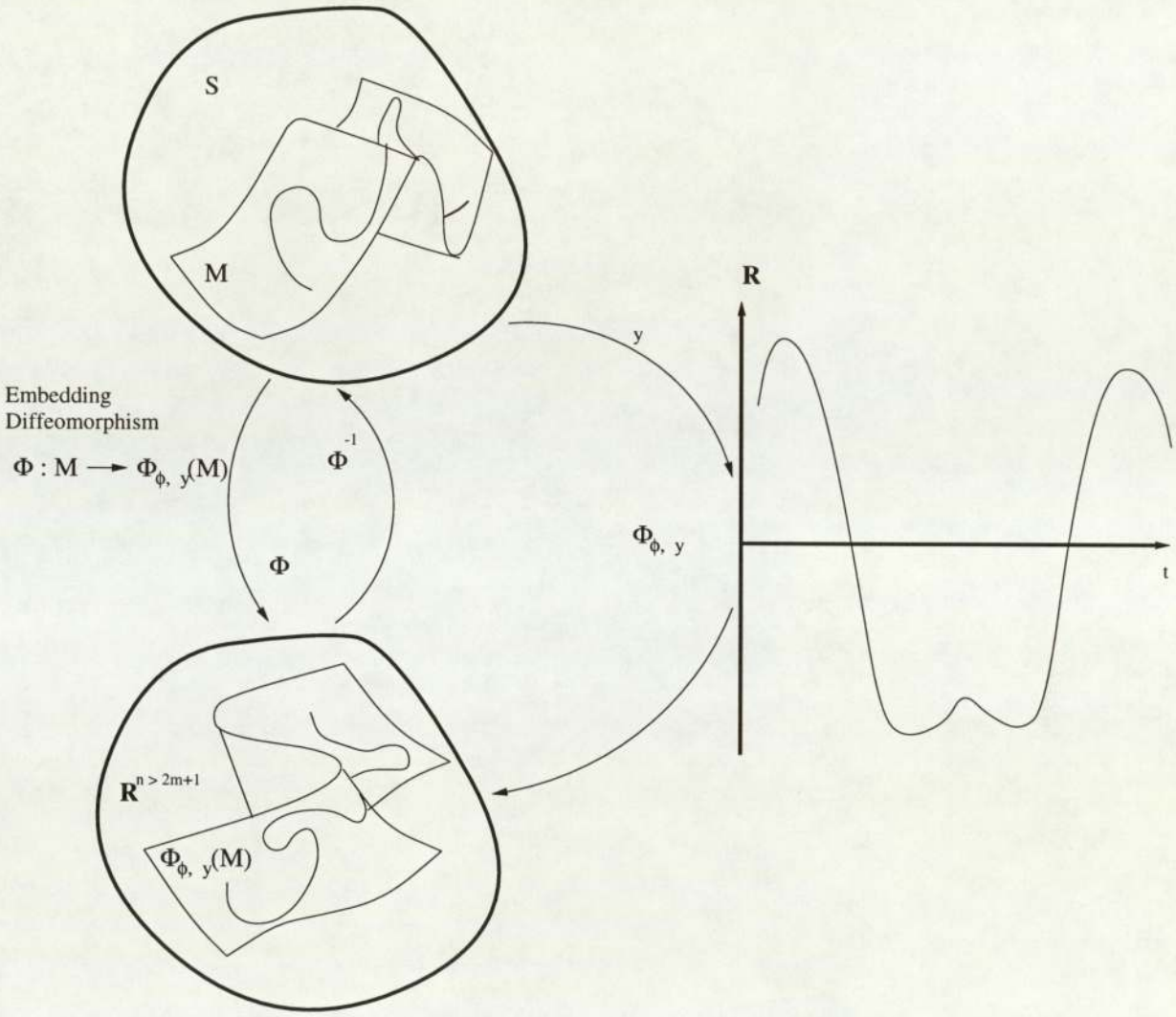


Figure 3.1: The embedding of a manifold into a Euclidean space with time series data using Takens' construction

$\varphi : \mathbf{M} \mapsto \mathbf{M}$ and $y : \mathbf{M} \mapsto \mathbb{R}$ satisfy the hypotheses of the embedding theorem, and m is chosen to be big enough for the map $\Phi_{(\varphi,y)}$ to be an embedding.

Given the choice we have made for y and φ, $\Phi_{(\varphi,y)}$ maps the system's state space to the observation's lag space

$$\begin{aligned}
\Phi_{(\varphi,y)}(\underline{\mathbf{x}}(t)) &= \big(y(\underline{\mathbf{x}}(t)), y(\varphi(\underline{\mathbf{x}}(t))), \ldots, y(\varphi^{2m}(\underline{\mathbf{x}}(t)))\big) \\
&= \big(y(\underline{\mathbf{x}}(t)), y(\underline{\mathbf{x}}(t+\tau)), \ldots, y(\underline{\mathbf{x}}(t+2m\tau))\big) \\
&= \big(x(t), x(t+\tau), \ldots, x(t+2m\tau)\big)
\end{aligned}$$

So, the trajectory of a 1-dimensional observation in a d-dimensional lag space and the trajectory of the system's state space differ only by a smooth local change of coordinates (as d is big enough). There are two main advantages to this. First, in a modelling context, there is no need for variables other than the lag values of the time series we are studying. Furthermore, the number of necessary lag values is directly linked to the number of the system's *degrees of freedom*, and this number can also be estimated by measuring statistics on the initial observation.

The strong relationship between the state space and the lag space allows us to manipulate lag values of an observation as if they were directly state variables of our system.

3.4 Degrees of Freedom

Given a time series $x(t)$, $t \in [\![1, N_{data}]\!]$, we have seen in the previous sections that it is possible to study the dynamics of the corresponding system by considering a sequence of $N = N_{data} - (n-1)$ vectors $\big(x(t), x(t+\tau), \ldots, x(t+(n-1)\tau)\big)$, $t = 1, \ldots, N$, where n is the *window size* of the embedding, τ is the *sampling interval*, and N_{data} is the

number of data points in the time series.

Thus, we can construct an $N \times n$ trajectory matrix $\mathbf{Z}$ from this sequence :

$$\mathbf{Z} = \begin{bmatrix} x(1) & x(1+\tau) & \dots & x(1+(n-1)\tau) \\ x(2) & x(2+\tau) & \dots & x(2+(n-1)\tau) \\ \vdots & \vdots & & \vdots \\ x(N) & x(N+\tau) & \dots & x(N+(n-1)\tau) \end{bmatrix}$$

As shown in [5], an analysis of the number of degrees of freedom in $\mathbf{Z}$ leads to a PCA of the trajectory matrix (For further explanations on PCA, see Section 2.1). After having constructed the embedding, a PCA is carried out on the transpose of this matrix. Using the notation of Section 2.1, $\{(\underline{\mathbf{e}}_1, \xi_1), \dots, (\underline{\mathbf{e}}_N, \xi_N)\}$ denotes the set of eigenvector/eigenvalue pairs of $\mathbf{cov}\{\mathbf{Z}\}$, with $\xi_1 \geq \xi_2 \geq \cdots \geq \xi_N$. We might think that, as the $\{\xi_i,\ i = 1, \dots, N\}$ are the root mean square projections of the trajectory matrix onto the basis vectors, the number of that are non-zero is the number of degrees of freedom. But $\mathbf{Z}$ is generally affected by experimental noise, which gives a wrong estimate of the number of degrees of freedom. The solution to this problem is to identify a *noise floor* in the eigenspectrum [5].

To find this noise floor, one of the simplest tests, and probably the most widely used, is the *scree test* [9] which derives from a visual inspection of the eigenvalues plotted against their root number. A typical eigenvalue plot shows a steep slope over the first few roots and a gradual trailing off for the rest of the roots (the scree). The term *scree* derives from the resemblance to the rubble that forms at the foot of a mountain. Cattell hypothesized the scree represented unwanted noise and that only the eigenvector/eigenvalue pairs prior to the scree should be retained for further use. The main problem with this method is the identification of the exact root where the scree begins to form, as it is rather subjective. An alternative to this method consists

of considering the log-eigenvalues plotted against their root number. This is the one we will adopt for our experiments. So, to sum up briefly, we will try to detect first major "kink" in the spectrum, and this will give us the number of degrees of freedom of the system.

3.5 Experimental Results

3.5.1 General comments about the data

The time series that has been used in our experiment is the raw time series formed from some British pound futures contract data. Even though the *open*, *high*, *low* and *close* fields were available, only the *close* field has been used. The original time series is composed of 4225 points, but we have only considered the first 1000 points. Given that a trading year is roughly made of 250 days, our data is therefore spread over 4 years. This signal is represented in Figure 3.2.

3.5.2 Determining the embedding delay τ

The main concern in the choice of the embedding delay τ is to avoid two major pitfalls. If one chooses too small a value for τ, then successive values are too close together and the embedding vectors are packed more or less around the identity line in the embedding space. On the other hand, when τ is too big, there is a big risk of losing information from all values between $x(t)$ and $x(t+\tau)$. So, the best thing possible in the choice of τ is to reach a trade-off. In this thesis, in order to incorporate every available data point, we have chosen $\tau = 1$.

3.5.3 Determining the window size of the embedding

The main problem with this part is that we are dealing with financial time series, which are well known to be nonstationary. Thus, we need to find a window size small enough

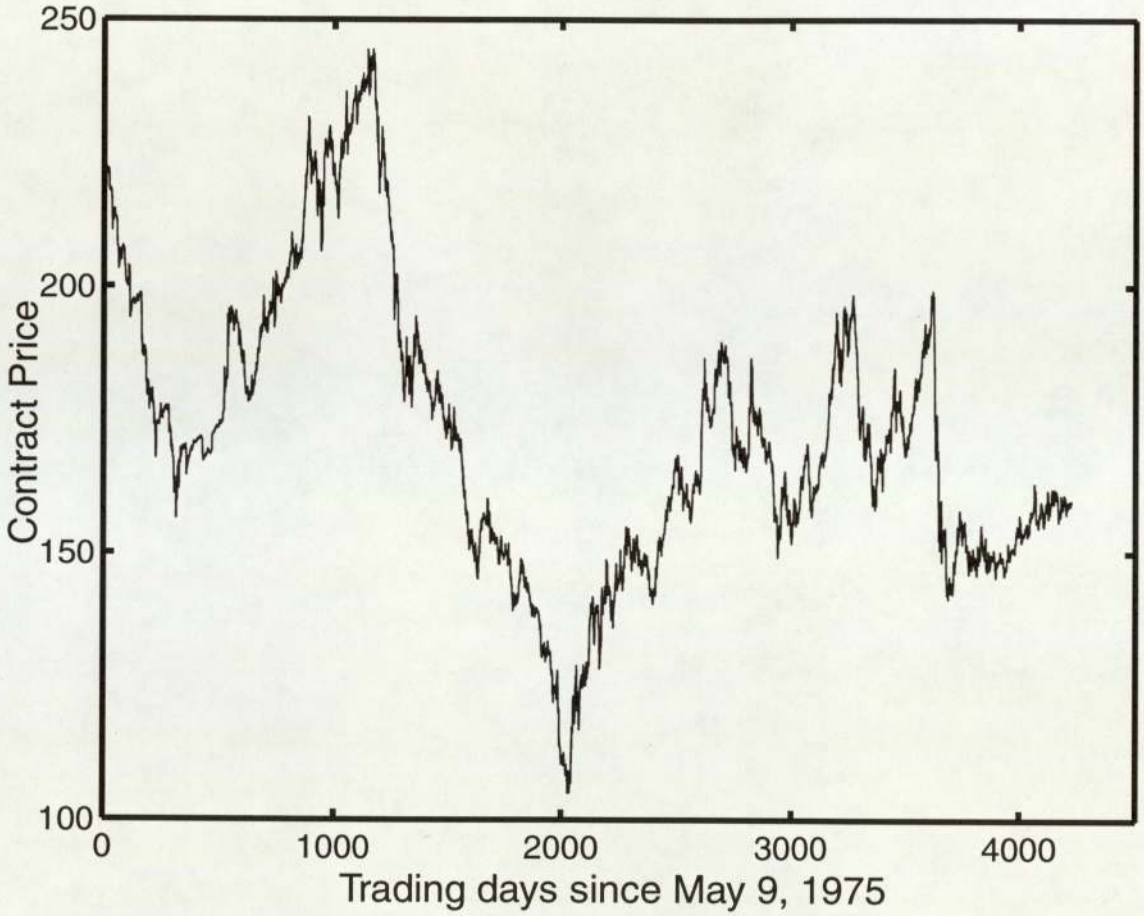


Figure 3.2: The original signal for the British pound futures contract data

so that the time series is stationary during this period. Finding such a window size is a very subjective problem. Thus, we cannot pretend to have an accurate approach to do it. Nevertheless, we can try to find the *best* window size for the embedding by looking at the eigenspectra for different window sizes. This is what can be seen in Figure 3.3. From this figure, we observe that there is a "convergence of the spectra" when the window size is of the order of 50, thus we chose $n = 50$.

3.5.4 Determining the degrees of freedom

Once the embedding parameters τ and n are set, it is interesting to determine the degrees of freedom of the data, by considering the singular spectrum of the delay vectors. The result of this procedure is shown on Figure 3.4. We know that for a signal free of noise, the spectra would be expected to show a smooth decline. Unfortunately, we can observe several kinks that may be attributed to the noise. Thus, by detecting the first major kink in the spectrum, we can have an idea of the actual complexity of the signal determistic component. Figure 3.4 shows that the the number of degrees of freedom is actually of order four.

We can then compute the percentage of variance explained by the first i dominant eigenvectors using the following formula:

$$\text{Percentage variance explained} = \frac{\sum_{j=1}^{i} \xi_j}{\sum_{j=1}^{n} \xi_j}$$

For $i = 4$, we find that the first four principal components explain 99.20% of the variance.

3.5.5 Shape of the attractor

Even if we do not have any idea about what the original manifold looks like, it is possible to study it using a diffeomorphism existing between the reconstructed space

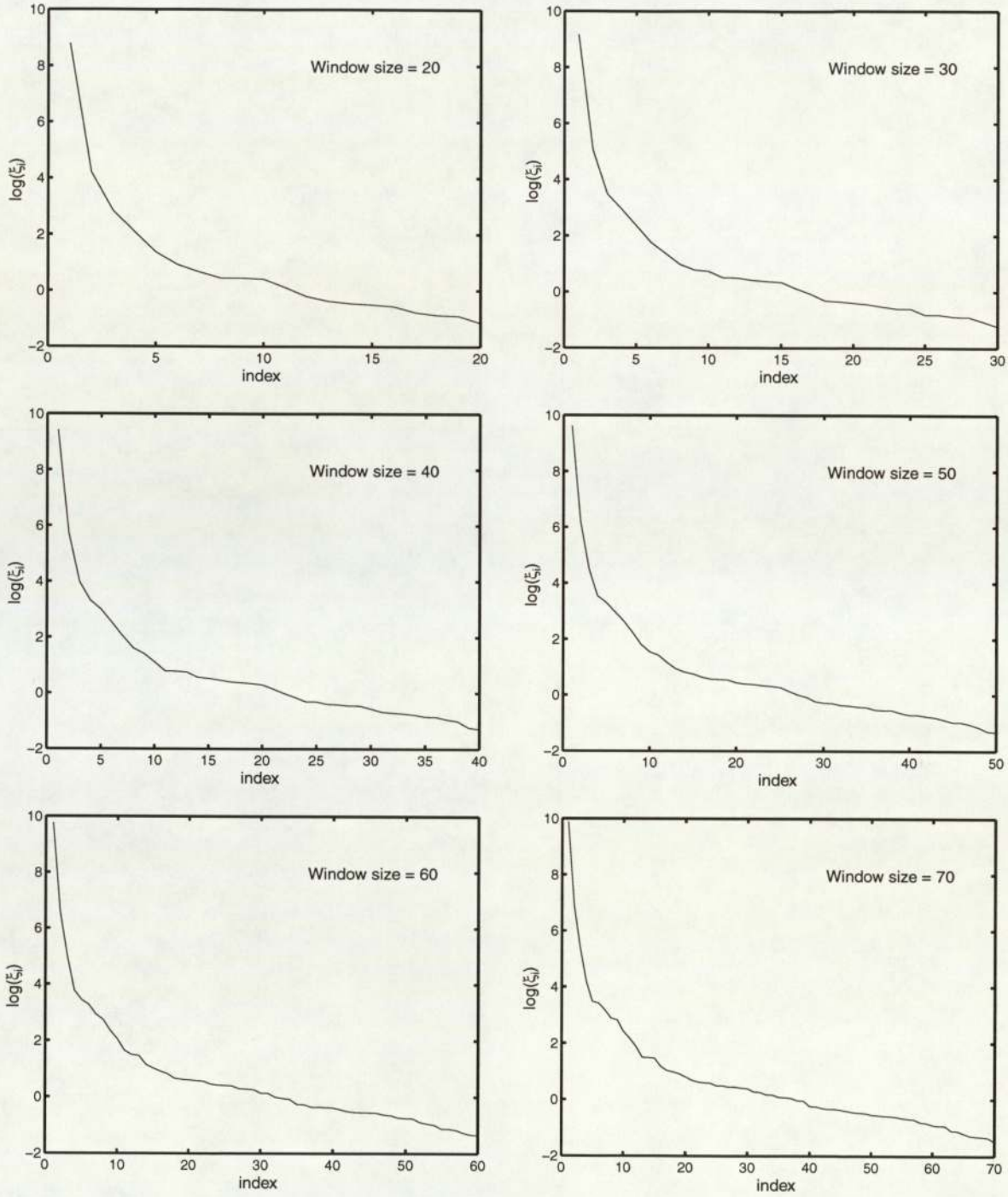


Figure 3.3: The eigenspectra for different window sizes for the British pound future contracts time series

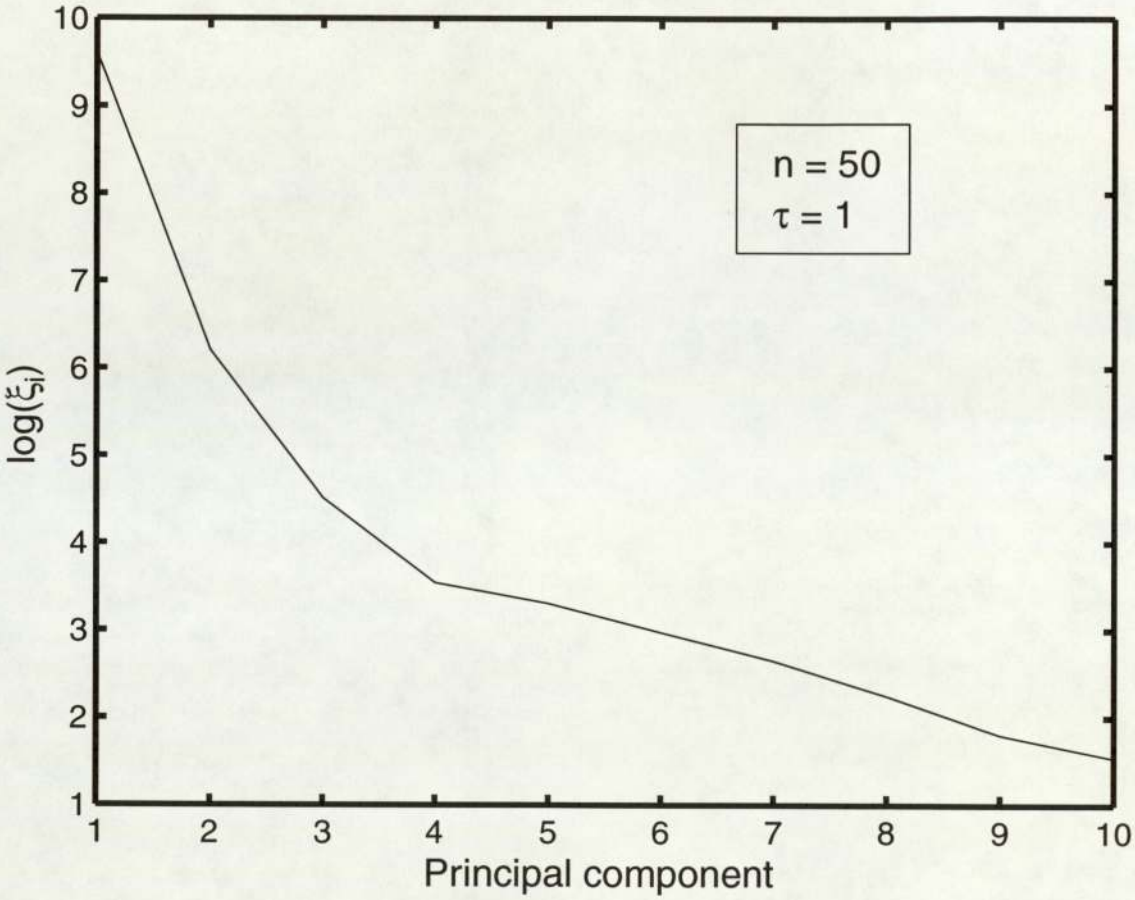


Figure 3.4: The eigenspectrum of the delay vectors restricted to the first ten principal components

and the original space. This is allowed because all transformations up to this point are linear, and do not create additional dynamics of their own. But we must be aware that four degrees of freedom is too high to visualize. That is why we will only study the projection of the data (i.e. the delay vectors) into the space spanned by the three dominant eigenvectors. For these eigenvectors, the percentage of variance explained is 98.98%.

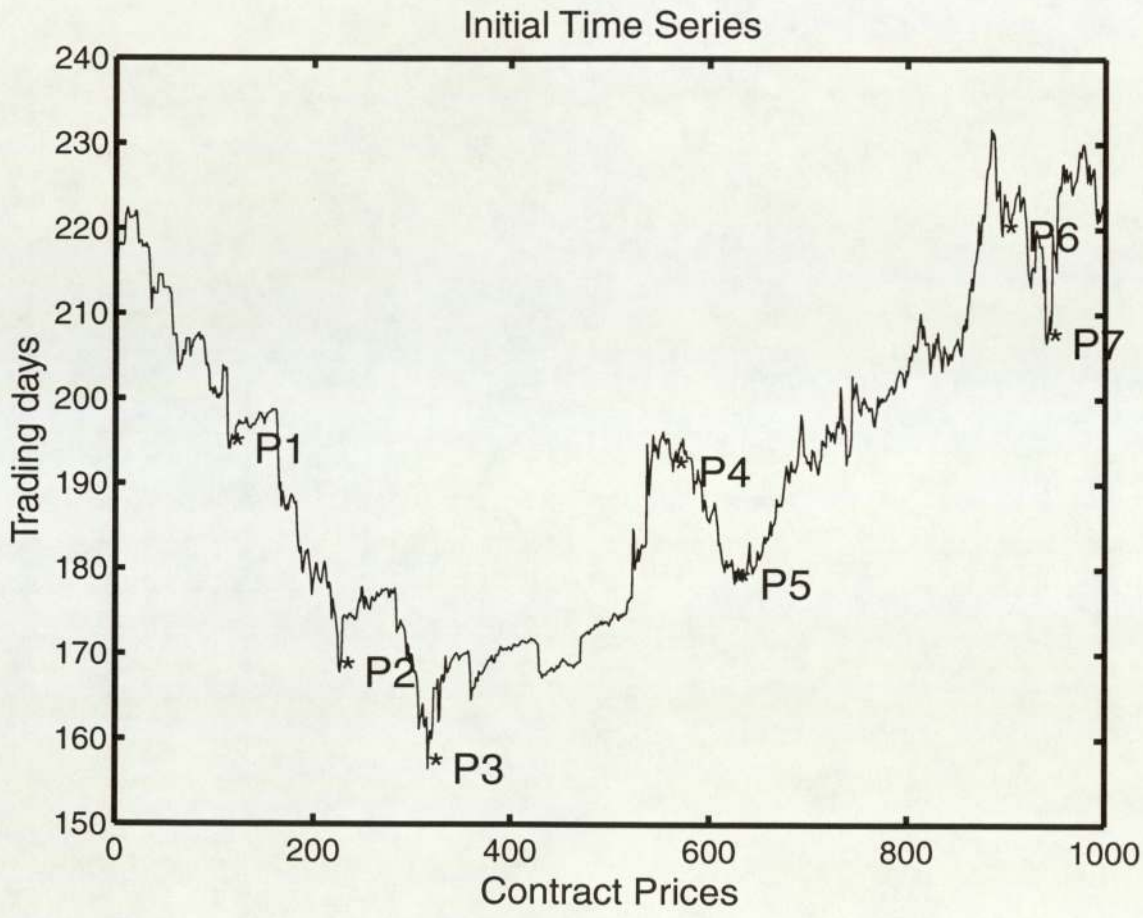


Figure 3.5: Initial Time Series

When looking at Figure 3.6, the first thing we can notice about the raw structure obtained by projecting the delay vectors into the three most significant directions (from the PCA point of view), is the shape of the structure. Indeed, we do not have a space filling figure, but a "bobbin-like" structure around which our signal is wrapped. Thus, with only four degrees of freedom, we have extracted some kind of geometric structure

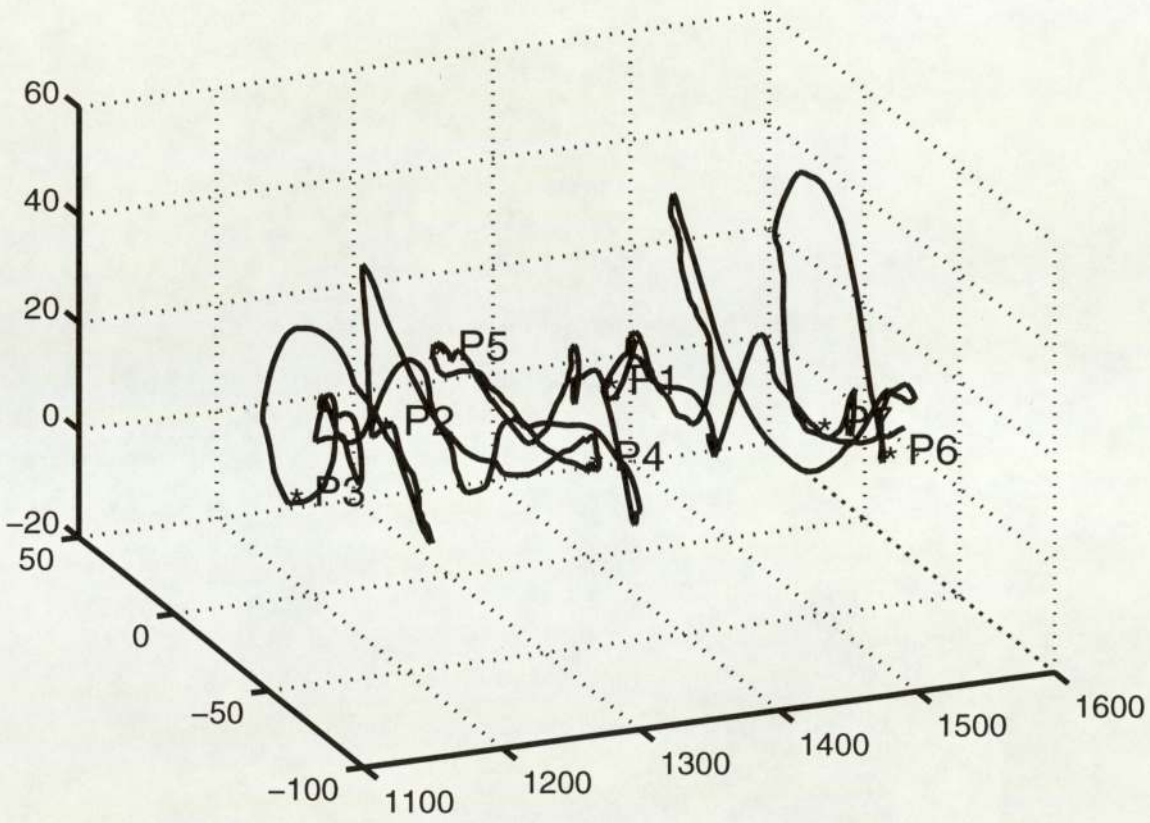


Figure 3.6: Projection of the delay vectors onto the three first eigenvectors

from the British pound data.

More interesting is the point that, when spotting some turning points on the original time series, we can study the evolution of this turning points in the "bobbin-like" structure (See Figures 3.5 and 3.6). It appears that every turning point of Figure 3.5 seems to match a change in direction concerning the structure of Figure 3.6.

But the problem with PCA is that it imposes such strong constraints as the orthogonality of axes, and only deals with the first and second order moments. In the event of independent sources in the series, such restrictions may lead to a rather poor performance of the PCA algorithm. As we shall see in the next chapter, ICA turns out to be the suitable tool for handling such situations.

Chapter 4

Independent Component Analysis

4.1 General Overview of ICA

Recently [3, 7, 24, 25, 29, 6, 16], blind source separation by Independent Component Analysis (ICA) has received attention because of its potential applications in signal processing such as in speech recognition systems, telecommunications and medical signal processing. The goal of ICA is to recover independent sources given only sensor observations that are unknown linear mixtures of the unobserved independent source signals. In contrast to correlation-based transformations such as Principal Component Analysis (PCA), ICA not only decorrelates the signal (using 2^{nd} order statistics) but also reduces higher-order statistical dependencies, attempting to make the signals *as independent as possible.* To sum up the main idea, one can say that ICA is a way of finding a linear non-orthogonal coordinate system in any multivariate data. The directions of the axes of this coordinate system are determined by both the second and higher order statistics of the original data. The goal is to perform a linear transform which makes the resulting variables *as statistically independent from each other as possible.*

Two different research communities have considered the analysis of independent components. On the one hand, the study of separating mixed sources observed in

an array of sensors has been a classical and difficult signal processing problem. The first studies undertaken by Herault and Jutten (1986) [15] have been further developed by Jutten and Herault (1991) [18], Karhunen and Joutsensalo (1994) [19], Cichoki, Unbehauen and Rummert (1994) [10]. Comon (1994) [11] elaborated the concept of independent component analysis and proposed cost functions related to the approximate minimization of mutual information between the sensors.

On the other hand, in parallel to blind source separation studies, unsupervised learning rules based on information-theory were proposed Linsker (1992) [22]. The goal of such methods was to maximise the mutual information between the inputs and outputs of a neural network. This approach is related to the principle of redundancy reduction suggested by Barlow (1961) [2] as a coding strategy in neurons. Each neuron should encode features that are as statistically independent as possible from other neurons over a natural ensemble of inputs. Bell and Sejnowski (1995) [3] also developed an algorithm turning the blind source separation problem into an information-theoretic framework and demonstrated the separation and deconvolution of mixed sources.

Other algorithms for performing ICA have been proposed, and we can try to classify the existing algorithms according to the following scheme :

	Method of solution		
Mathematical approach	Diagonalization	Fixed point	Gradient method
Fourth order cumulants	`JADE`	Original fixed point	Bigradient
Contrasts based on other nonlinearities	—	Generalized fixed point	Bigradient

Table 4.1: A classification of ICA algorithms.

The `JADE` algorithm is the only one used in our experiments, nevertheless we will

describe briefly all these algorithms.

4.2 Entropy Maximization

Though it is possible to perform ICA by maximizing the entropy of the outputs in a two-layer neural network, it is very difficult to form a computational procedure based directly on the general theory, mainly because we cannot determine the density functions of the unknown components. To overcome this problem, Bell and Sejnowski [3] have derived a gradient ascent algorithm for changing the weights of the first layer (linear layer) and the biases of the second layer to maximize the entropy of the outputs. They use the *logistic* transfer function

$$g(x) = \frac{1}{1 + e^{\omega x + \omega_0}}$$

where ω controls the steepness of the function and ω_0 is a bias parameter. The performance of this algorithm depends on how well the nonlinear transfer functions approximate the cumulative distribution functions of the independent components. The main limit to this algorithm is that it may not work for negatively kurtotic sources, as suggested by Bell and Sejnowski.

4.3 Fixed-Point Algorithms

This algorithm has been introduced by Hyvärinen and Oja [16]. Basically, the idea is to optimize a contrast function. We have already seen that kurtosis can be used as an optimization criterion for ICA. Let us consider the optimization of kurtosis of the projection of a zero mean whitened random variable $\underline{\mathbf{v}}$ onto vector $\underline{\mathbf{w}}$

$$f(\underline{\mathbf{w}}) = \mathbf{E}\left[(\underline{\mathbf{w}}^T \underline{\mathbf{v}})^4\right] - 3\,\|\underline{\mathbf{w}}\|^4$$

under the constraint

$$h(\underline{\mathbf{w}}) = \|\underline{\mathbf{w}}\|^2 - 1 = 0$$

Using the method of Lagrange multipliers, we find the following necessary condition for an optimum

$$4\mathbf{E}\left[(\underline{\mathbf{w}}^T\underline{\mathbf{v}})^3\underline{\mathbf{v}}\right] - 12\,\|\underline{\mathbf{w}}\|^2\underline{\mathbf{w}} + 2\lambda\underline{\mathbf{w}} = \underline{\mathbf{0}}$$

which can be rewritten as

$$\lambda'\underline{\mathbf{w}} = \mathbf{E}\left[(\underline{\mathbf{w}}^T\underline{\mathbf{v}})^3\underline{\mathbf{v}}\right] - 3\underline{\mathbf{w}}$$

using $\|\underline{\mathbf{w}}\|^2 = 1$ and $\lambda' = -\frac{1}{2}\lambda$. This equation show that at an optimum $\underline{\mathbf{w}}$ the right hand side of the equation is parallel with $\underline{\mathbf{w}}$, or that the direction of $\underline{\mathbf{w}}$ remains fixed under the iteration

$$\underline{\mathbf{w}}(k+1) = \mathbf{E}\left[(\underline{\mathbf{w}}(k)^T\underline{\mathbf{v}})^3\underline{\mathbf{v}}\right] - 3\underline{\mathbf{w}}(k)$$

The fixed point algorithm is guaranteed to find one source, starting from a random initial point.

If one needs to find more sources, it is possible to use extra information from the fact that the whitened ICA basis vectors are orthogonal, ie by orthogonalizing the found vector $\underline{\mathbf{w}}$ against the subspace spanned by previously found vectors. The main problem with this algorithm is due to the termination criterion: instead of having a termination point being the fixed point of the iteration, we have it as the fixed point of the iteration-orthogonalization-normalization combination.

Finally, there is an alternative to this algorithm using a simultaneous search for all basis vectors of ICA as well as a symmetric orthogonalization [16].

4.4 The Bigradient Algorithm

Let us suppose that the kurtoses of the sources have the same sign. Then we can find a new contrast function

$$\begin{aligned} J_{\underline{\mathbf{x}},\sum\kappa}(B) &= \sum_{i=1}^{n} \kappa_4(\underline{\mathbf{b}}_i^T \underline{\mathbf{x}}) \\ &= \sum_{i=1}^{n} \mathbf{E}\left[(\underline{\mathbf{b}}_i^T \underline{\mathbf{x}})^4\right] - 3\left(\mathbf{E}\left[(\underline{\mathbf{b}}_i^T \underline{\mathbf{x}})^2\right]\right)^2 \end{aligned}$$

where $B = [\underline{\mathbf{b}}_1 \dots \underline{\mathbf{b}}_n]^T$ and κ_4 denotes the fourth-order cumulant, ie the kurtosis. Using the fact that $\mathbf{E}\left[(\underline{\mathbf{b}}_i^T \underline{\mathbf{v}})^2\right] = 1 \quad \forall i \in [\![1, n]\!]$ for any whitened random vector $\underline{\mathbf{v}}$, we can drop out the second term of the equation and then rewrite a new simplified objective function

$$J_{\underline{\mathbf{v}},\sum}(B) = \sum_{i=1}^{n} \mathbf{E}\left[(\underline{\mathbf{b}}_i^T \underline{\mathbf{x}})^4\right] \tag{4.1}$$

To find the ICA basis, we should optimize (4.1) under the constraint "B is orthogonal". To perform this, we can use a penalty function approach, where the penalized object function is

$$J_{\underline{\mathbf{v}},penalty}(B) = \underbrace{J_{\underline{\mathbf{v}},\sum}(B)}_{original\ objective} + \rho \underbrace{\|I - B^T B\|_F^2}_{orthogonalization} \tag{4.2}$$

and $\rho \in \mathbb{R}$ is a penalty coefficient and $\|X\|_F = \sum_{i=1}^{n}\sum_{j=1}^{n} {x_{ij}}^2$ denotes the *Frobenius norm* of matrix X.

The actual optimization is performed using a gradient descent algorithm which yields in the final form of the bigradient algorithm

$$B(k+1) = B(k) + \mu_k \mathbf{E}\left[\underline{\mathbf{v}}(k)(\underline{\mathbf{v}}(k)^T B(k))^3\right] + \gamma_k B(k)[I - B(k)^T B(k)]$$

where μ_k is an adaptation parameter which is either a small constant or decreases slowly with the number of iterations, and acceptable values of γ_k ($\gamma_k = \mu_k \rho_k$) have been observed to lie in the range $[0.5, 1]$. One can notice that, like in the fixed point algorithm, the third power of the algorithm may be changed to another nonlinearity

g[1], thus yielding the generalized bigradient algorithm

$$B(k+1) = B(k) + \mu_k \mathbf{E}\left[\underline{\mathbf{v}}(k) g(\underline{\mathbf{v}}(k)^T B(k))\right] + \gamma_k B(k)[I - B(k)^T B(k)]$$

Nevertheless, this algorithm is only able to separate sources all of which have either positive or negative kurtoses.

4.5 Joint Approximate Diagonalization of Eigenmatrices

4.5.1 The algorithm

The JADE algorithm of Cardoso and Souloumiac is based on joint approximate diagonalization of eigenmatrices. In Section 2.5, we have seen that the ICA problem could be solved by computing the eigenvectors of the cumulant matrix $Q_{\underline{\mathbf{v}}}(M)$ for any matrix M. JADE is an extension of this idea in that it diagonalizes a set of eigenmatrices representing the whole cumulant matrix set $C_{\underline{\mathbf{v}}}$. The diagonalization in JADE proceeds by maximization of

$$C(P) = \sum_{M \in C_{\underline{\mathbf{v}},e}} \|\mathrm{diag}(P^T M P)\|^2$$

where P is an orthogonal matrix and $C_{\underline{\mathbf{v}},e}$ is a set of eigenmatrices of $Q_{\underline{\mathbf{v}}}$ with nonzero eigenvalues

$$C_{\underline{\mathbf{v}},e} = \{\xi M \mid Q_{\underline{\mathbf{v}}}(M) = \xi M \wedge \xi \neq 0\}$$

(Here $\mathrm{diag}(P^T M P)$ is a vector formed by scanning the diagonal elements of $P^T M P$). In [8], it has been shown that $|C_{\underline{\mathbf{v}}}| = n$, so the computations need only be performed over a set of n matrices. For the ICA model, the optimization over this eigenset $C_{\underline{\mathbf{v}},e}$ is equivalent to the optimization over the whole set $C_{\underline{\mathbf{v}}}$, thus giving a significant computational advantage.

[1] $g(t) = \tanh(t)$ is quite popular for such a function

As described in [8], the JADE algorithm can be described as a 4-step process :

Step 1 : Form the sample covariance matrix $\hat{R}_x$ and compute a whitening matrix $\hat{W}$.

Step 2 : Form the 4^{th} order cumulants $\hat{Q}_z$ of the whitened process $\hat{z}(t) = \hat{W}x(t)$; Compute the n most significant eigenpairs $\{\hat{\xi}_r, \hat{M}_r \mid 1 \leq r \leq n\}$.

Step 3 : Jointly diagonalize the set $\{\hat{\xi}_r \hat{M}_r \mid 1 \leq r \leq n\}$ by a unitary matrix $\hat{U}$.

Step 4 : An estimate of A is $\hat{A} = \hat{W}^{\#}\hat{U}$

Step 1 concerns the 2^{nd} order statistics part, and is implemented via eigendecomposition of $\hat{R}_x$. According to the white noise assumption, an estimate of the noise variance $\hat{\sigma}$ is the average of the $m - n$ smallest eigenvalues of $\hat{R}_x$. Let $\mu_1, \ldots, \mu_n$ be the n largest eigenvalues and $\underline{\mathbf{h}}_1, \ldots, \underline{\mathbf{h}}_n$ the corresponding eigenvectors of $\hat{R}_x$. Thus, a whitener $\hat{W}$ is $\hat{W} = \left[\frac{1}{\sqrt{\mu_1 - \hat{\sigma}}}\underline{\mathbf{h}}_1, \ldots, \frac{1}{\sqrt{\mu_n - \hat{\sigma}}}\underline{\mathbf{h}}_n\right]^T$.

In step 2, computation of the eigenmatrices amounts to diagonalizing a $n^2 \times n^2$ matrix made from the elements of $\hat{Q}_z$. As we deal with *real-valued signals*, it is possible to exploit the symmetries of the cumulants to further reduce the number of matrices to be diagonalized. Furthermore, it is not necessary to compute the eigenmatrices as it suffices to jointly diagonalize a set of cumulant matrices.

Step 3 is implemented using a variant of the single-matrix Jacobi technique to several matrices, called Givens rotations. (See Appendix A for further explanations.)

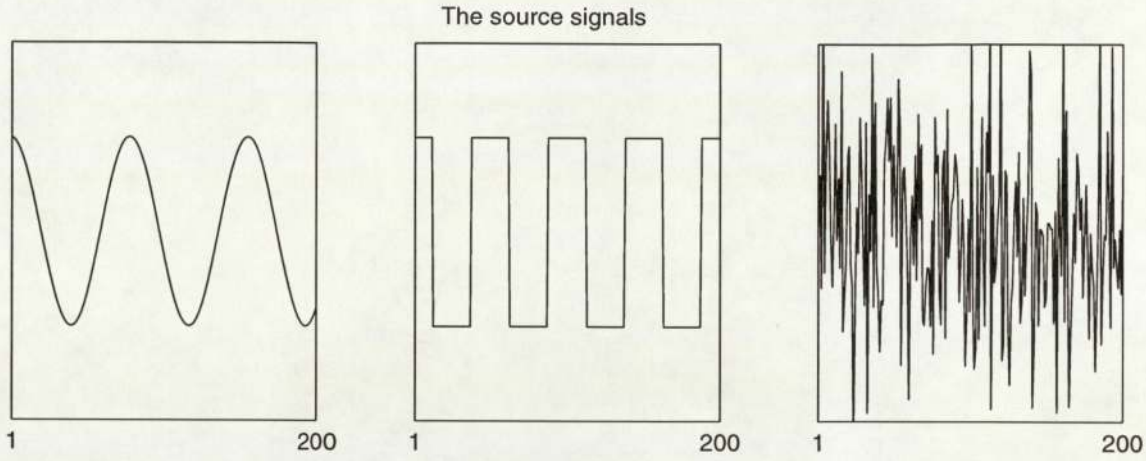
In step 4, the pseudo-inverse of $\hat{W}$ need not be explicitly computed: the eigendecomposition of $\hat{R}_x$ may be recycled by $\hat{W}^{\#} = \left[\sqrt{\mu_1 - \hat{\sigma}}\underline{\mathbf{h}}_1, \ldots, \sqrt{\mu_n - \hat{\sigma}}\underline{\mathbf{h}}_n\right]$.

4.5.2 A toy problem

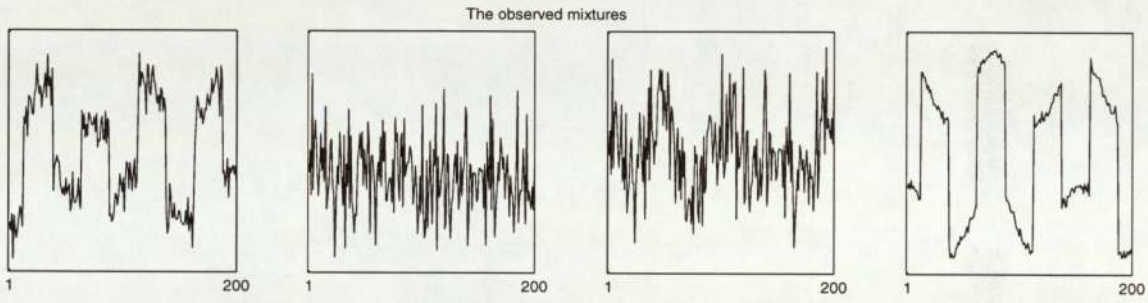
An example of how the JADE algorithm performs can be seen by considering the following toy problem. Let s_1, s_2 and s_3 denote the three independent sources defined by

- s_1 : A sine wave
- s_2 : A square wave
- s_3 : A white Gaussian process

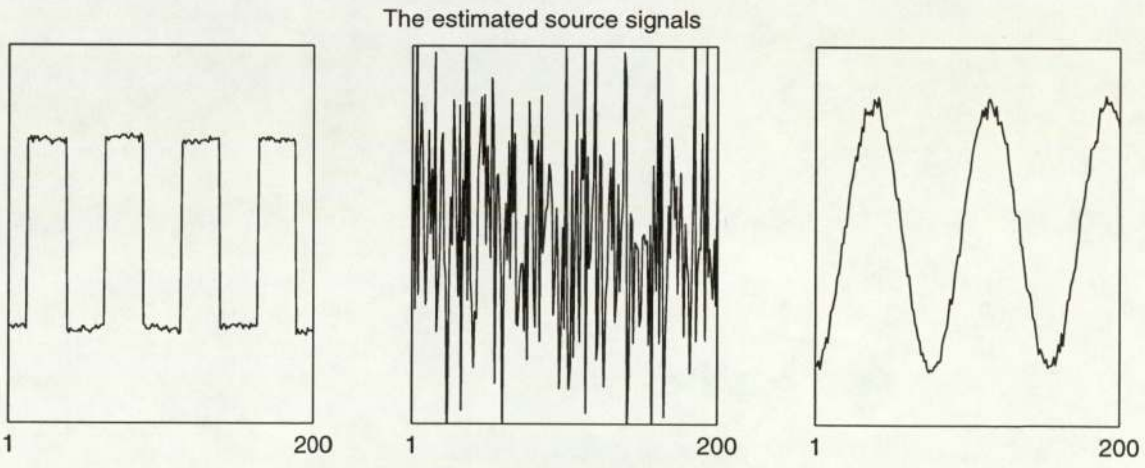
We just mix these signals by a 4×3 random matrix, and then add a small Gaussian noise to the resulting mixture. Then, we run the JADE algorithm on the mixtures to extract the new recovered signals. The three steps of this procedure are shown in Figure 4.1



(a) Source signals



(b) Mixture signals+noise



(c) Recovered signals

Figure 4.1: An example of how the JADE algorithm performs on a mixture of signals

Chapter 5

ICA Results

Previous Chapters have discussed the ability of ICA to detect and extract independent sources from the time series. This is both interesting and powerful, and presents a great interest for financial times series that are very often the resultant of sometimes drastically independent factors (traders' actions, natural catastrophe, governmental decisions, etc).

5.1 General Information Concerning ICA

5.1.1 Reasons to use ICA in finance

We have seen previously in Chapter 4 that ICA provides a mechanism of decomposing a given signal into statistically independent components. Thus, it might seem interesting to explore whether ICA can give some indications of the underlying structure of the stock market, by finding a bunch of interpretable factors. Such factors can be economic indicators (unemployment, internal product, ...), business-related information (assets, debt, productivity, type of production, announcements), monetary parameters, interest rates, relevant political events (international wars, government directives, ...). Most often, it is very difficult to find which of these variables are relevant and how to evaluate their effect without the help of a human expert.

5.1.2 Description of the data

As one of our main concerns in using an ICA algorithm, the observed signals need to be stationary. Thus, we use the first difference $\nabla x(t) = x(t) - x(t-1)$ instead of the raw data $x(t)$. This is the input of our embedding process, with $\tau = 1$, $n = 50$. The embedding matrix is now the entry of our ICA algorithm , that is the `JADE` algorithm [8] (See Section 4.5).

5.1.3 Structure of the independent components

In all our experiments concerning financial data, we assume that the number of signals provided as an input to the algorithm equals the number of sources supplied to the mixing model. As mentioned by Cardoso in the code of `JADE`, there is a practical limit to the number of independent components that can be extracted with this implementation, and we reach this limit using $n = 50$. As all the ICA computations are quite time-consuming, we have therefore reduced the number of data points available to the first 500 data points of our time series. As we have quite a large number of sources, each time we plot some results, only a few of them are represented.

We can see a few examples of what these independent components look like in Figure 5.1.

The plots shown in Figure 5.1 reveal two very interesting features. It has already been mentioned in Section 2.3 that the `JADE` algorithm can only recover independent sources most often permuted and rescaled. For instance, components IC_1, IC_2 and IC_3 seem very similar (taking in consideration this scaling problem). Thus, this suggests that there might be some kind of clustering among the independent components. This issue will be discussed later on in this thesis. Another problem we have to be aware of in this chapter is the absence of order amongst the independent components, or more precisely, how to find an accurate way to order the recovered sources.

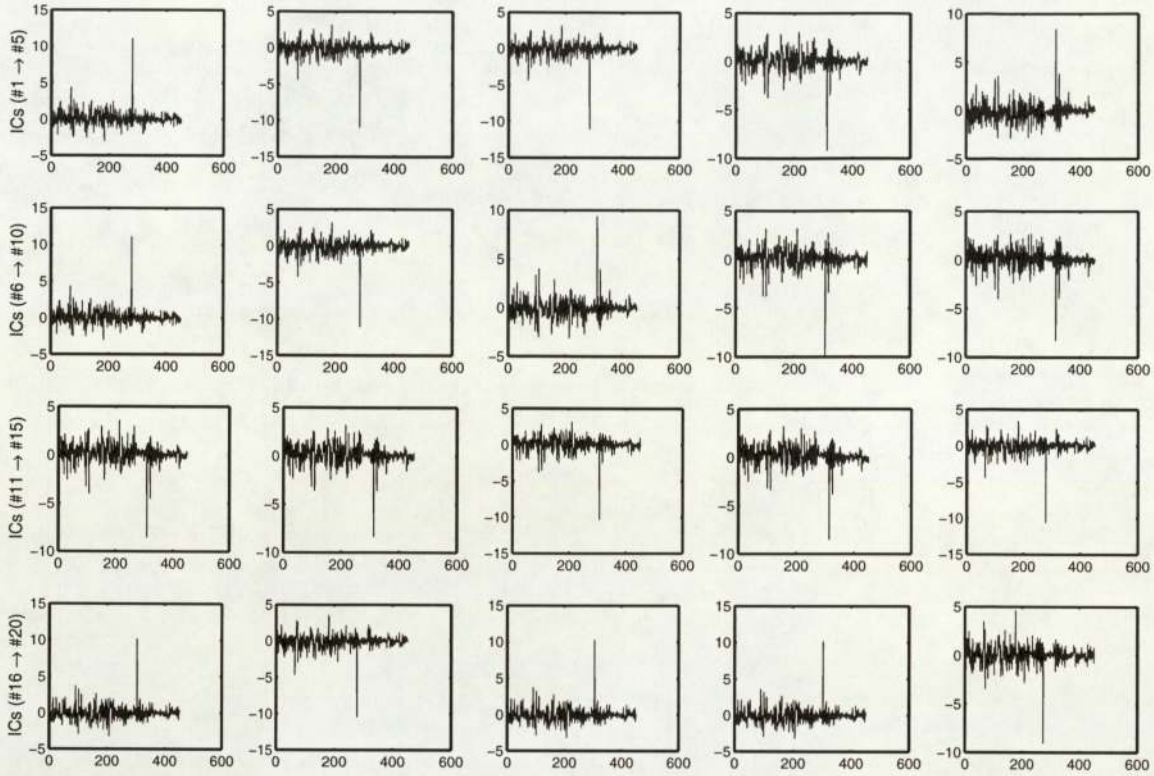


Figure 5.1: The first 20 independent components recovered by the JADE algorithm

5.2 Reconstruction of the Time Series

One of the main concerns of this section is to reconstruct an approximation to the initial time series using only a few independent components.

5.2.1 Reconstruction algorithm

Using equation 2.1 of Section 2.3, we can rewrite it as :

$$x_i(t) = \sum_{j=1}^{n} a_{ij} s_j(t) \tag{5.1}$$

Thus it is possible to obtain the reconstruction of the i^{th} return in terms of the estimated ICs as

$$\hat{x}_i(t-j) = \sum_{k=1}^{n} a_{ik} y_k(t-j) \quad j = 0, \ldots, N-1 \tag{5.2}$$

where $y_k(t-j)$ is the value of the k^{th} estimated IC at time $t-j$ and a_{ik} is the weight in the i^{th} row, k^{th} column of the estimated mixing matrix A. A is obtained as the inverse

of the demixing matrix W. Then, the reconstructed prices are found as

$$\begin{cases} \hat{p}_i(j+1) &= \hat{p}_i(j) + \hat{x}_i(j) \quad j = t-N, \ldots, t-1 \\ \hat{p}_i(t-N) &= p_i(t-N) \end{cases}$$

One can notice that using the whole set of ICs (that is 50 ICs) allows us to reconstruct the original time series, that is, there is no information lost in any stage of the ICA process. This is due to the fact that the product of the mixing matrix and the independent sources matrix is the starting matrix ($X = AS$).

As we have already mentioned before, ranking the ICs is a difficult task. Furthermore, there are various criteria to sort them. In their implementation of the JADE algorithm, Cardoso and Souloumiac [8] used an Euclidean norm to sort the rows of the demixing matrix W according to their contribution across all signals. In [1], Back and Weigend used an L_∞ norm on the weighted ICs to show the ICs which cause the maximum price change in a particular stock. We used a different way of sorting the ICs as explained in the following paragraph.

5.2.2 Sorting the independent components

The method presented here is inspired by the following phenomenon. We want to rank the ICs by order of importance in the reconstruction process, so we consider $\tilde{\mathbf{Z}}_i$, the approximate value of the embedding matrix obtained by considering IC number i. Then we construct matrix $\mathbf{S_i}$ whose i^{th} line is made of the i^{th} independent source, and all other terms are zero, that is

$$\mathbf{S_i} = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ s_{i1} & s_{i2} & \ldots & s_{iN} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix}$$

By constructing $\tilde{\mathbf{Z}}_i = A\mathbf{S}_\mathbf{i}, \quad i = 1, \ldots, n$, we obtain a new set of n approximations to the embedding matrix. We then compute $\|\tilde{\mathbf{Z}}_i - \mathbf{Z}\|$, the error in the approximation of $\mathbf{Z}$ by $\tilde{\mathbf{Z}}_i$. Therefore, we can sort these values in increasing order, such that the index corresponding to the minimum error value gives us the most important IC, the one corresponding to the second minimum error, the second most important IC, and so on.

For the following step, we sort the columns of the mixing matrix and the rows of the independent sources matrix according to this order, and then follow the procedure described in Section 5.2.1.

Figure 5.2 shows all 50 different curves obtained by using 1 to 50 ICs in the reconstruction process.

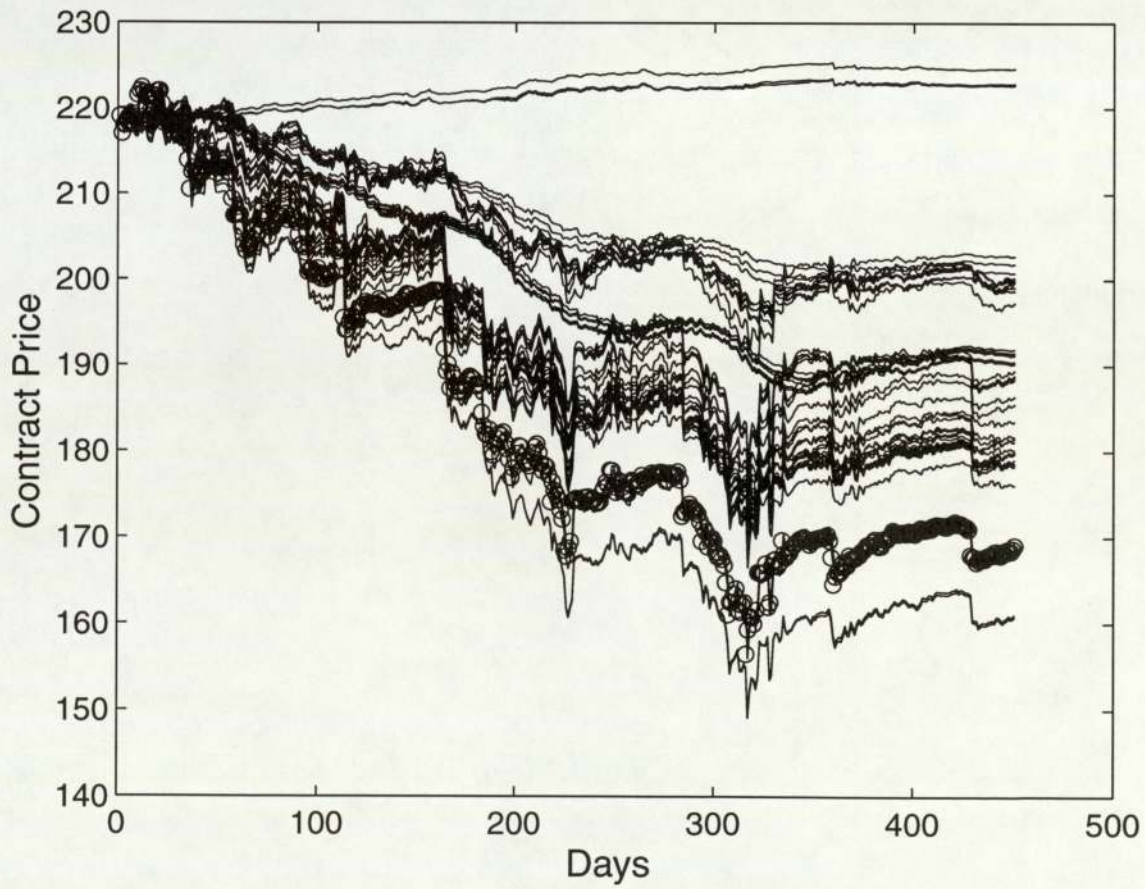


Figure 5.2: Reconstruction of the Contract Price time series using the i most important ICs, $i = 1, \ldots, 50$. The actual data are indicated by open circles.

5.2.3 Interpretation of the results

Even though Figure 5.2 is a bit overloaded, because of the superimposition of all the plots, it is reassuring to see that the original time series (represented with circles) is perfectly reconstructed when using the whole set of independent components, as mentioned in Section 5.2.1.

While analyzing the figure, the most obvious feature we can see is a kind of clustering in the reconstructed time series, more particularly 6 different groups of curves. These groups are isolated in Figure 5.3 to ease their identification.

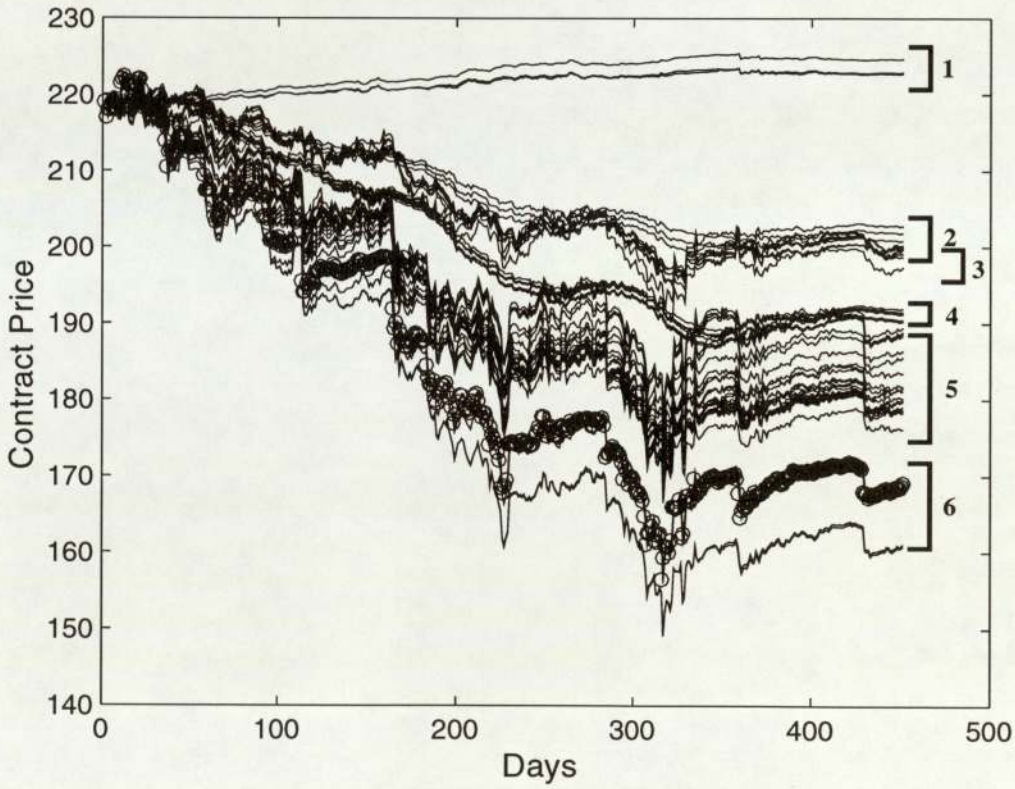


Figure 5.3: Illustration of the clustering phenomenon

These clusters suggest that in each group, there is no need to consider several ICs, as the reconstruction process should not be affected by adding several independent components belonging to the same cluster. If we carefully attempt to identify this

cluster, a "by hand" approach reveals the following clusters :

Cluster 1	$IC_1 \rightarrow IC_3$
Cluster 2	$IC_4 \rightarrow IC_6$
Cluster 3	$IC_7 \rightarrow IC_{13}$
Cluster 4	$IC_{14} \rightarrow IC_{22}$
Cluster 5	$IC_{23} \rightarrow IC_{45}$
Cluster 6	$IC_{46} \rightarrow IC_{50}$

These clusters are identified by vertical lines on pictures 5.4 and 5.5.

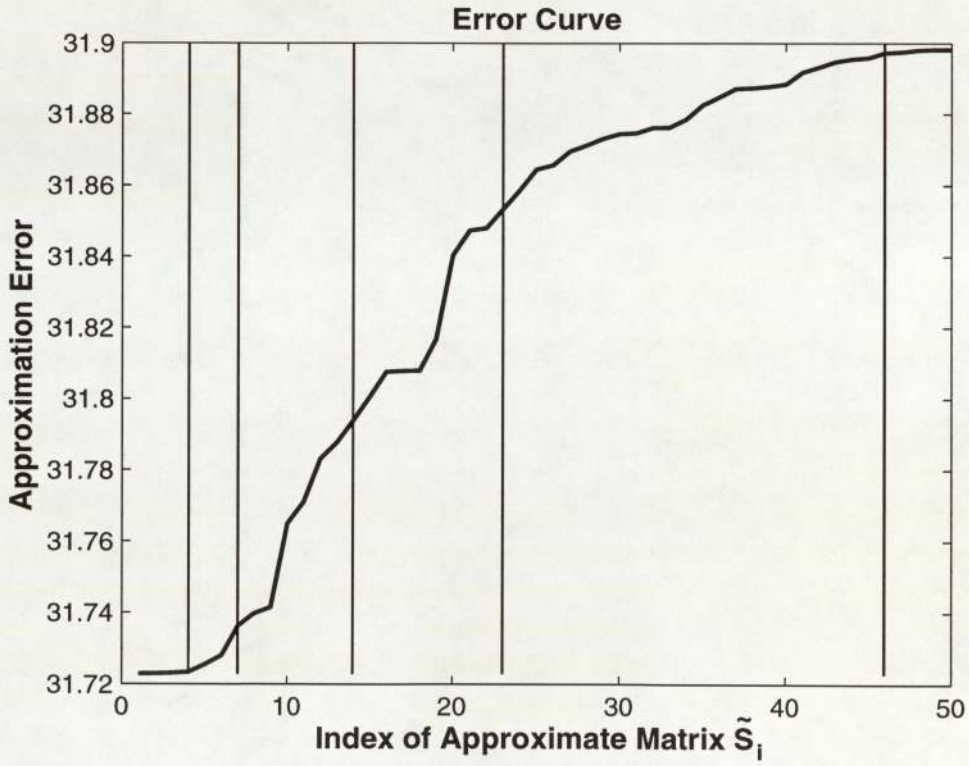


Figure 5.4: Cluster identification. The ICs are sorted in increasing order of the norm magnitude $\|\tilde{\mathbf{Z}}_i - \mathbf{Z}\|$.

In the following paragraph, we will see how the reconstruction is affected by taking just the first IC of each cluster.

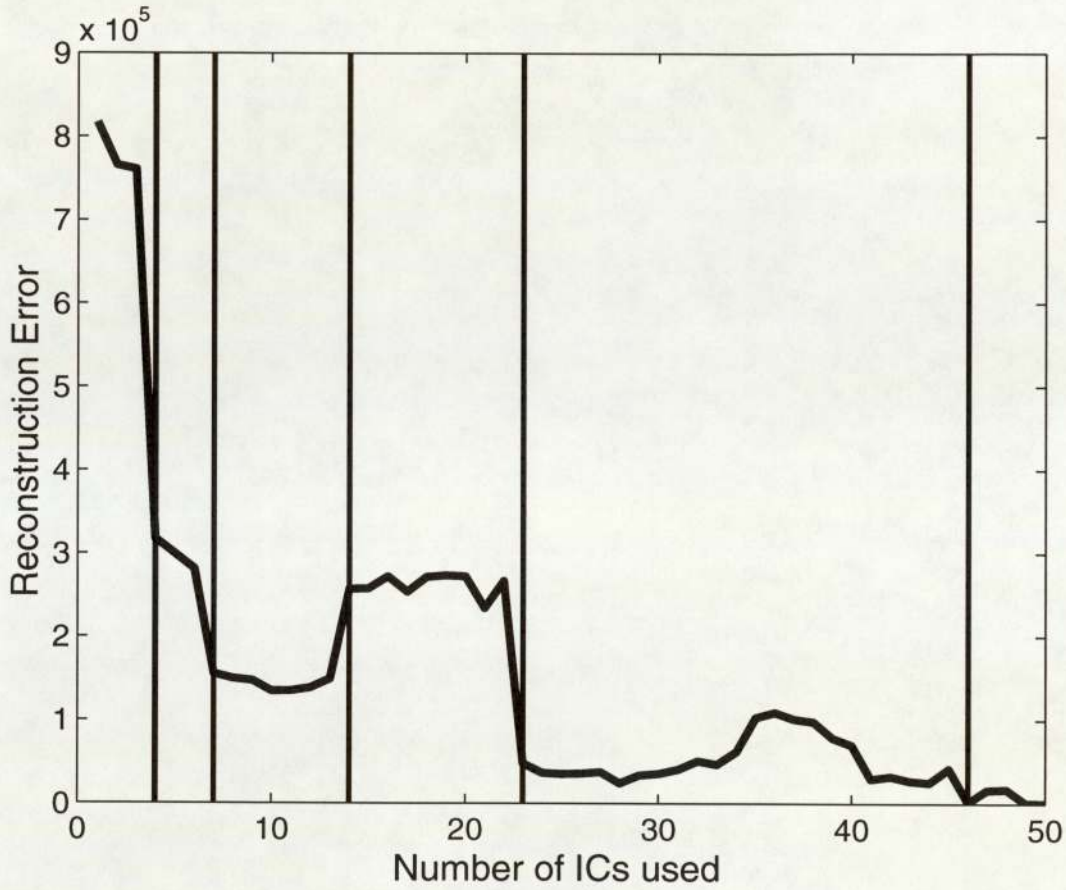


Figure 5.5: Cluster identification. The reconstruction error is computed using the Mean-Square Error between the original time series and its approximant

5.2.4 Reconstruction using clusters

In this part, we see how the reconstruction is affected by considering only one component of each cluster described in the previous section. To simplify this procedure, we have chosen to consider only the first component of each cluster, that is IC_1, IC_4, IC_7, IC_{14}, IC_{23} and IC_{46}. The new reconstruction curves are shown in Figure 5.6, whereas the new reconstruction error is plotted in Figure 5.7.

Even if we can see that the reconstruction process improved a lot after clustering, it is interesting to look carefully at Figure 5.7. Indeed, it suggests that IC_{14} – that is, the first IC of cluster 4 – does not contribute to the signal, as it increases the reconstruction error. Thus, we can try to reproduce this experiment by dropping out IC_{14} from the reconstruction. The new results are shown in Figures 5.8 and 5.9. Now, the reconstruction error curve is monotonic decreasing.

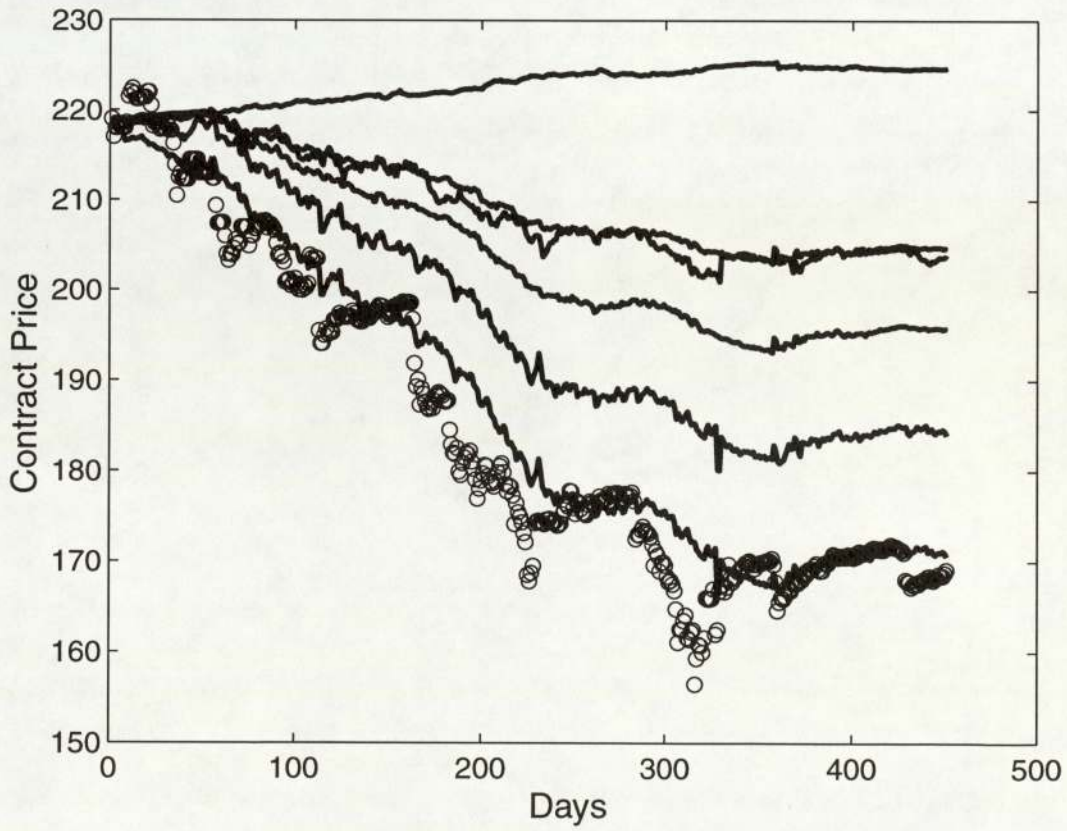


Figure 5.6: Reconstructed Time Series using 1 to 6 ICs

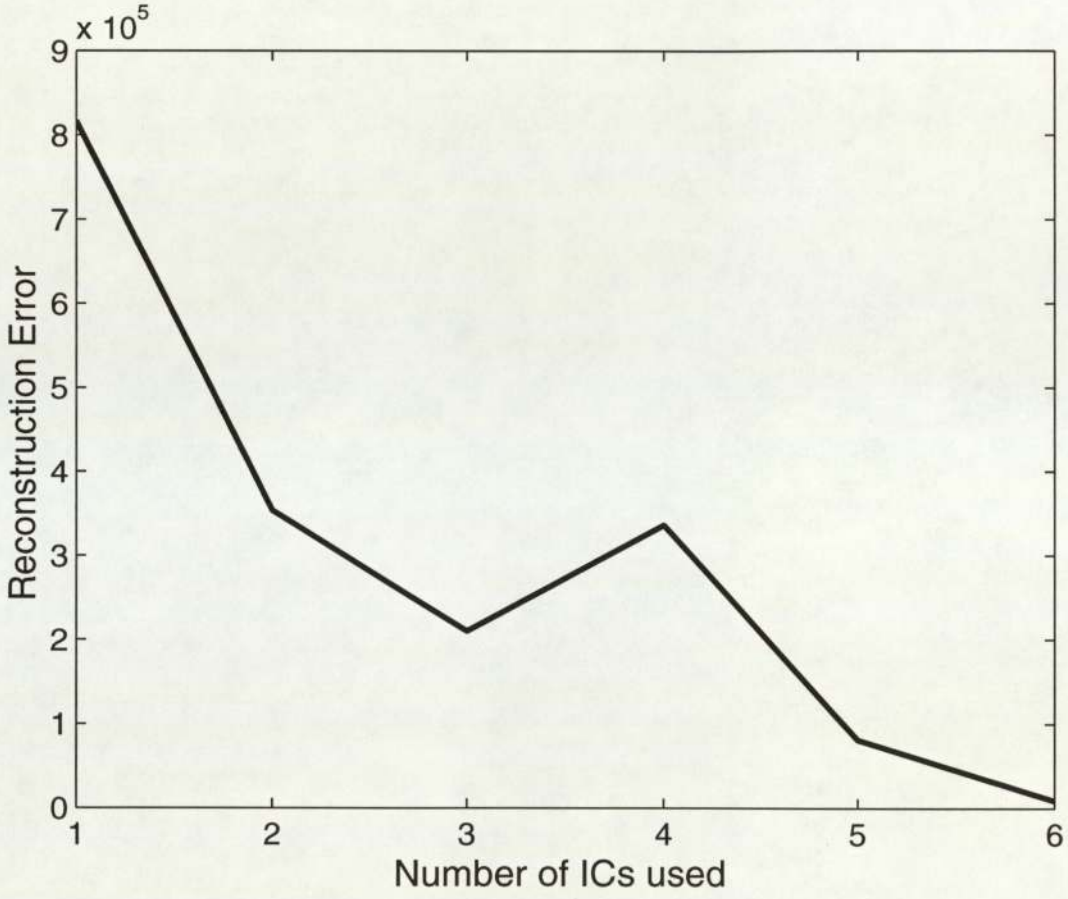


Figure 5.7: Reconstruction Error after Clustering

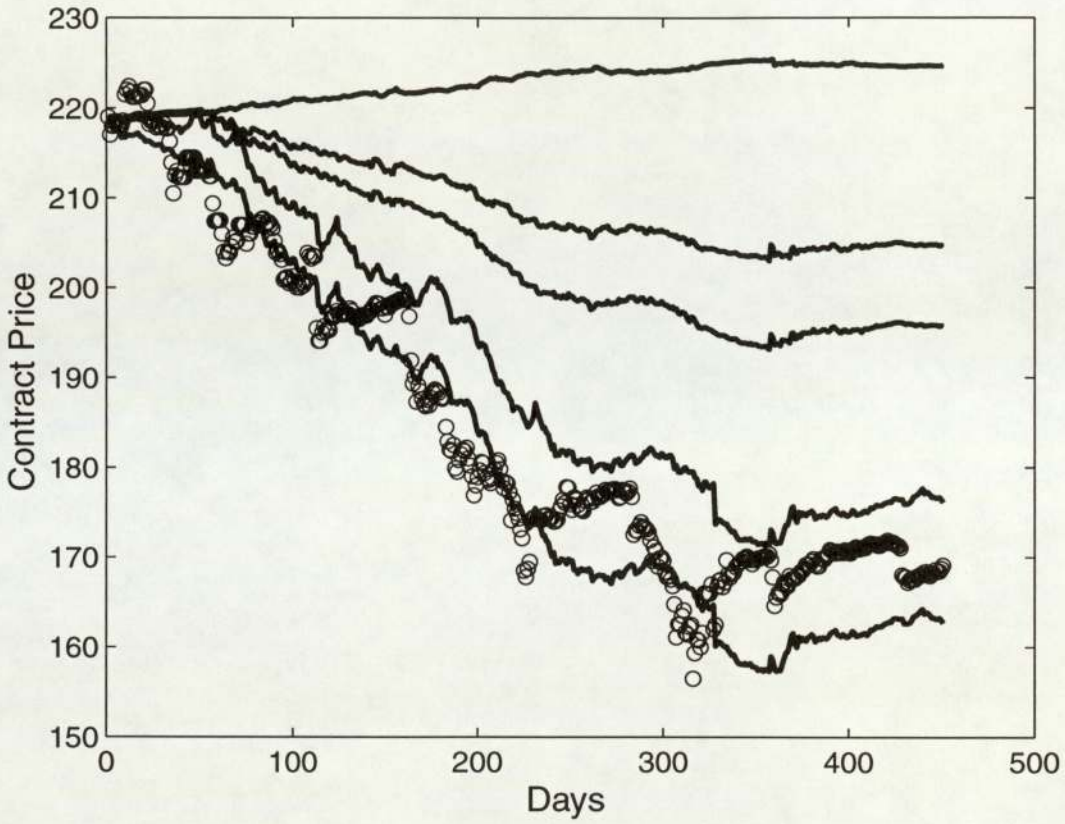


Figure 5.8: Reconstructed Time Series using 1 to 5 ICs

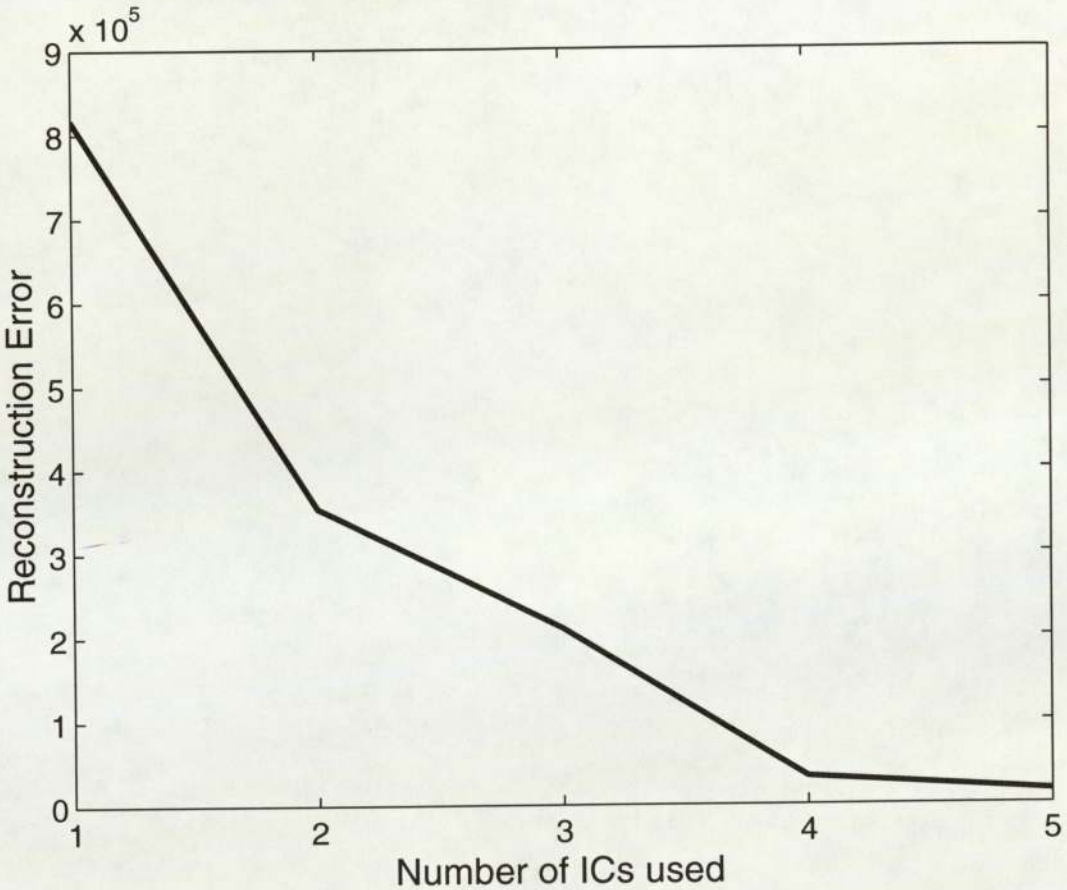


Figure 5.9: Reconstruction Error after Clustering (without IC_{14})

5.3 Thresholded Reconstruction

In the previous section, we have discussed the effect of reconstructing the original time series by considering the cumulative sums of only a few dominant ICs. This section goes further by thresholding these dominant ICs. The threshold process is simple : if a weighted IC value is below a threshold, we set it to zero. Thus, only the values above the threshold are used to reconstruct the signal.

The thresholded reconstructions slightly differ from the normal reconstruction, as we can see in equations 5.3 and 5.4. The new algorithm is presented below :

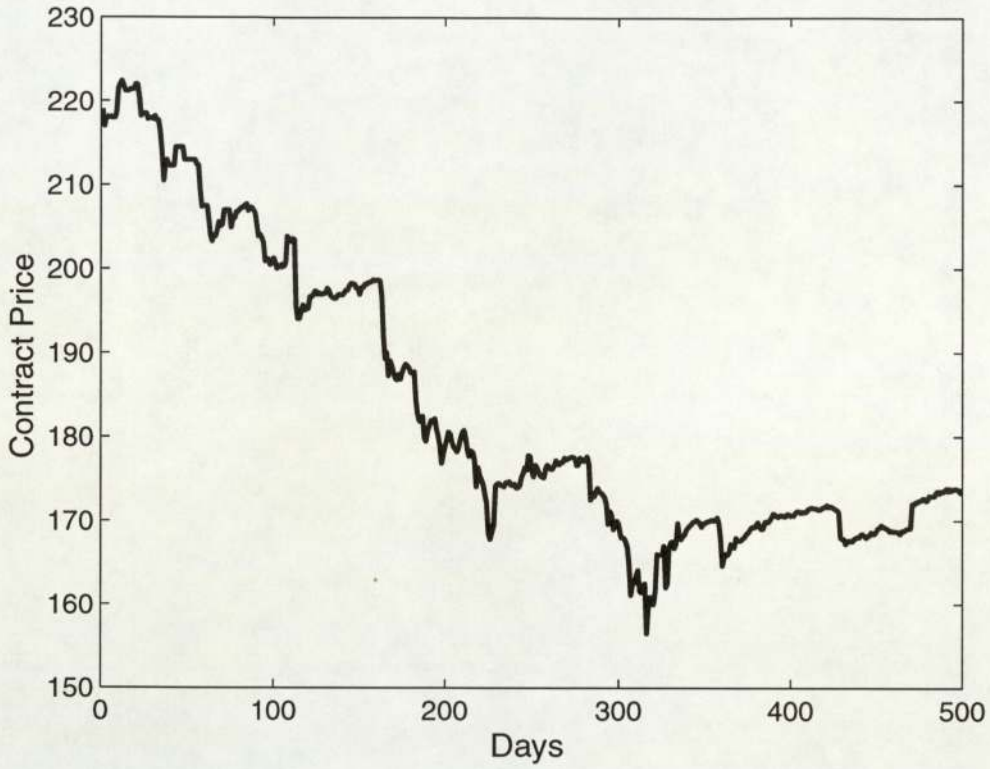
$$\begin{aligned} \bar{x}_i(t-j) &= \sum_{k=1}^{n} g(\underbrace{a_{ik}y_k(t-j)}_{weighted\ ICs}) \quad j=0,\ldots,N-1 \\ &= \sum_{k=1}^{n} g(\bar{y}_{ik}(t-j)) \quad j=0,\ldots,N-1 \end{aligned} \tag{5.3}$$

$$g(u) = \begin{cases} u & |u| \geq \varepsilon \\ 0 & |u| < \varepsilon \end{cases} \tag{5.4}$$

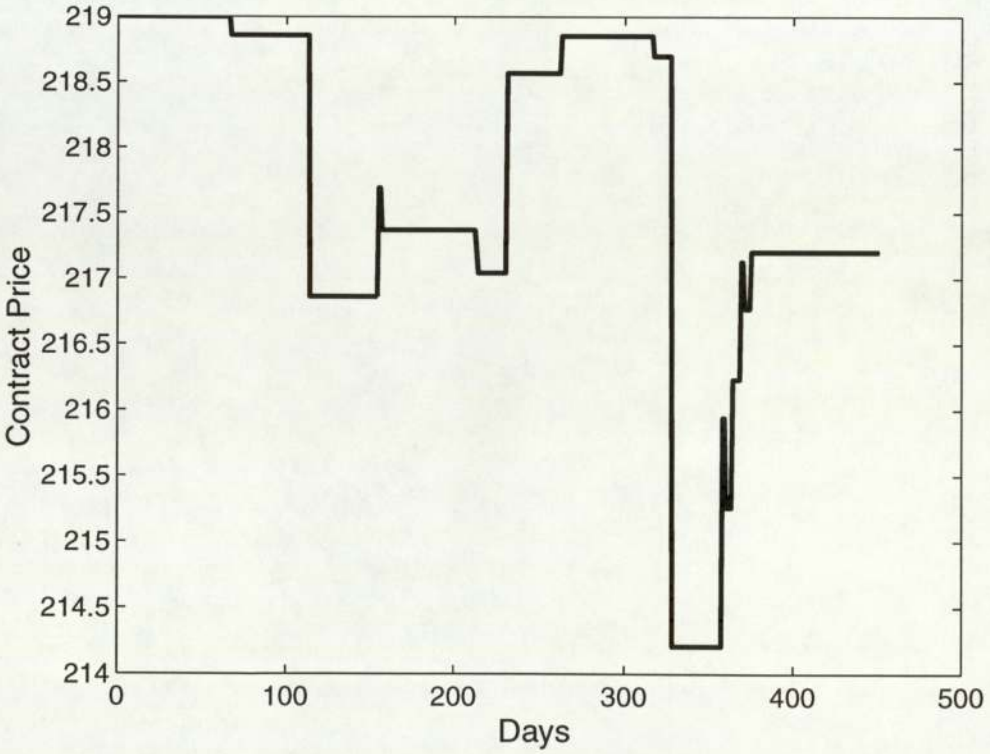
where $\bar{x}_i(t-j)$ denotes the reconstructed returns using thresholds, $g(.)$ is the threshold function and ε is the threshold value. The threshold was set arbitrarily to a value, such that most of the lower level components were excluded. The new reconstructed prices – after thresholding – are found as

$$\begin{cases} \hat{p}_i(j+1) &= \hat{p}_i(j) + \bar{x}_i(j) \quad j=t-N,\ldots,t-1 \\ \hat{p}_i(t-N) &= p_i(t-N) \end{cases}$$

The contract price reconstructed from the thresholded returns are shown in Figure 5.10 b. This figure indicates that the thresholded ICs provide useful information, as well as a tool to extract the turning points of the original time series (Figure 5.10 a).



(a) Original Time Series



(b) Thresholded Time Series

Figure 5.10: Reconstructed prices obtained by computing the cumulative sum of only the thresholded values of the five most dominant ICs. The original time series at the top is just plotted for comparison purposes.

5.4 Comparison with PCA

As we discussed earlier (See Chapter 2), PCA is a well established statistical method of data analysis. In this section, we seek to compare the performance of PCA with ICA.

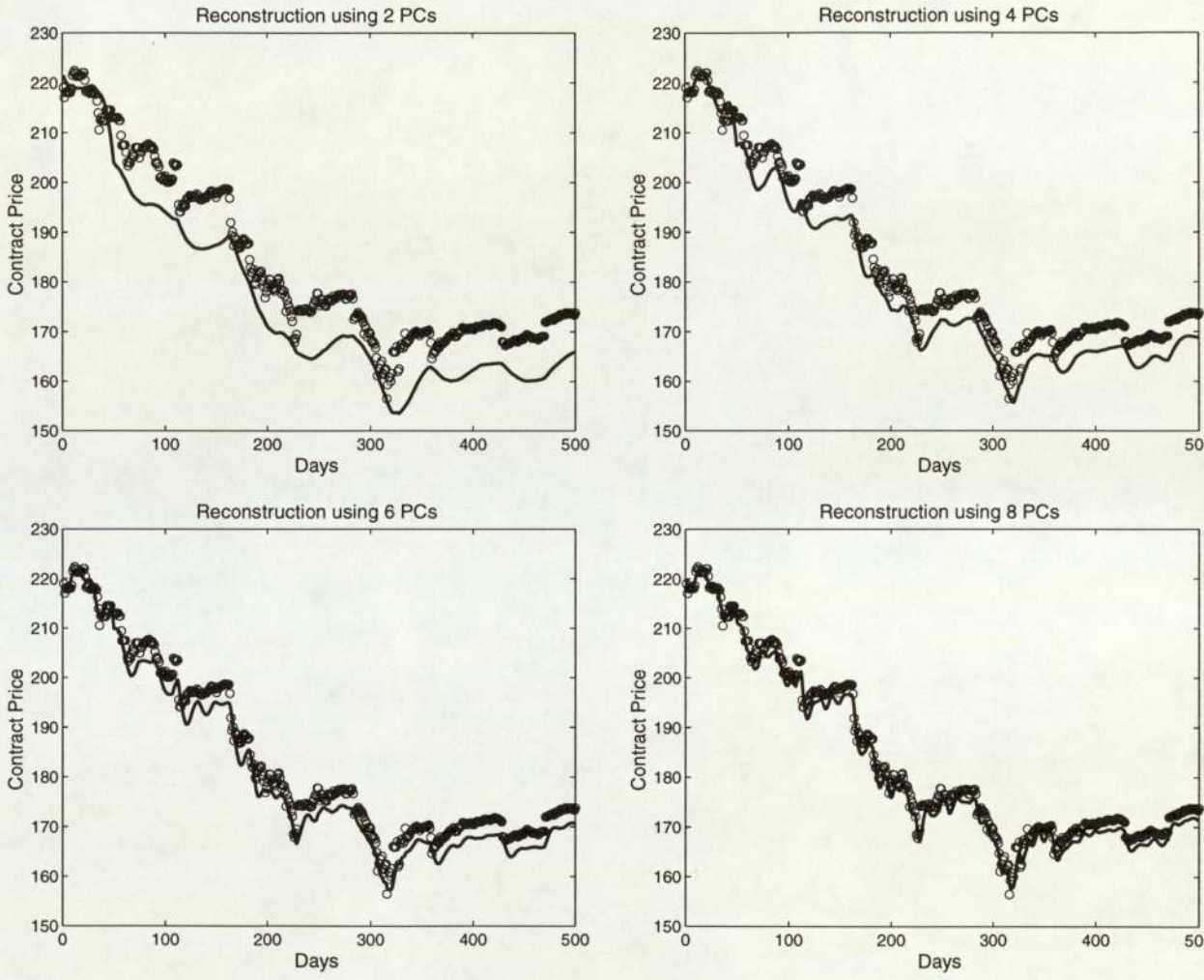


Figure 5.11: Reconstructed prices obtained by computing the cumulative sum of the 2, 4, 6 and 8 first PCs, respectively. The original time series is plotted with open circles.

From the above results, it turns out that PCA yields a better performance in the reconstruction of the original time series.

Chapter 6

Conclusions

6.1 Dynamical Embedding

Throughout this thesis, we have dealt with a single univariate time series. Data analysis techniques – such as ICA – are most often used in a multi-channel context, so we first need to preprocess our data. To address this problem, we have used a technique called embedology, or more precisely a class of embedding methods known as *Takens' Delay Coordinate Maps* [5, 33, 34]. The strength of this method is that it allows us to study the dynamical properties of a system without knowing any information about the original manifold. Furthermore, even if the *attractor* is disfigured, all the topological properties are preserved. Finally, it is the basic step for any further study in our thesis. The main difficulty encountered for this part is due to the very subjective approach in the way of finding the embedding parameters – such as the window size or the time delay – to provide a time interval over which the time series is quasi stationary. When we have used the scree test, it could have been possible to consider other methods [30].

6.2 ICA

The ICA approach we have pursued in this thesis is quite novel. Indeed, people generally address the problem of multi-channel ICA. Furthermore, unlike EEG signals

analysis, image processing and signal processing, for which there has already been a widespread applications of ICA, there had so far been only few papers concerning the application of ICA to financial markets [1]. So, it was really challenging to investigate the field of single-channel ICA applied to financial time series. We have implemented an efficient way to rank the ICs, which is not an easy task, as one of ICA weaknesses is the absence of order among the ICs. Our experiments have also revealed evidence of clustering. All this information has been used in order to reconstruct an approximation of the original time series with only a few ICs. We have also considered that by thresholding those few ICs, it is possible to get some morphological information about the time series.

6.3 Limitations of the ICA model

Our main concern about the `JADE` algorithm is that it is a *batch* algorithm. Thus, it cannot address the problem of too much data. Moreover, we have used it close to its limits, that is with a number of sources *too high*. This resulted in incredibly long running time sessions, which suggested that another algorithm might be more appropriate.

6.4 Conclusion

The theory of embedology provides a powerful framework for financial time series analysis and preprocessing. The use of Takens' delay coordinate maps along with PCA and ICA, has allowed the reconstruction of an unseen attractor based solely on observations of a single time series – a British Pound future contract.

6.5 Future Work

Throughout this thesis, the majority of our experiments were solely based on Futures Contract data. In order to assess the real performance of the methods used, it would be sensible to test them on data exhibiting different behaviour such as Commodities data.

Another study which would be of great interest is how to find a means to detect clusters automatically. Indeed, our study revealed that the independent components are grouped by clusters. But we have only found these clusters "by hand", which is far from an optimal method to do it. Although some kind of k-means clustering was briefly attempted during the course of the research, we can think of techniques as Kohonen's Self-Organising Map (SOM) [20], or Sammon Mapping [31] to be more appropriate, while keeping in mind that such a technique should provide us with a way to identify which components belong to a given cluster.

Concerning the ICA point of view, in all our experiments, we used the `JADE` algorithm [8]. The main reason that influenced this decision is that this algorithm is originated from research in signal processing, and rests on strong foundations. Furthermore, it has already been used by Back and Weigend [1] in a financial data context. But we have also seen that it is used close to its limits, so it might be interesting to test another algorithm such as the *fast fixed-point algorithm* designed by the ICA group, from Helsinki University of Technology [16].

Eventually, we can think about other techniques such as NLPCA [21], Curvilinear Component Analysis [13], or context-sensitive ICA [28] as possible extensions to the work undertaken in this thesis.

References

[1] A. Back and A. Weigend. A first application of independent component analysis to extracting structure from stock returns. *International Journal of Neural Systems*, 8:473–484, 1997.

[2] H. Barlow. Possible principles underlying the transformations of sensory messages. pages 217–234. MIT Press, Cambridge, MA, 1961.

[3] A. J. Bell and T. J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[4] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1996.

[5] D. S. Broomhead and G. P. King. Extracting qualitative dynamics from experimental data. *Physica*, 20D:217–236, 1986.

[6] J.-F. Cardoso. Infomax and maximum likelihood for blind separation. In *IEEE Signal Processing Letters*, volume 4, pages 112–114, 1997.

[7] J.-F. Cardoso and B. Laheld. Equivarient adaptative source separation. In *IEEE Transactions on Signal Processing*, volume 45, pages 434–444, 1996.

[8] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non-Gaussian Signals. *IEE Proc F*, 140(6):362–370, 1993.

[9] R. Cattell. The scree test for the number of factors. *J. Multiv. Behav.*, 1:245–276, 1966.

[10] A. Cichocki, R. Unbehauen, and E. Rummert. Robust learning algorithm for blind separation of signals. *Electronics Letters*, 30(17):1386–1387, 1994.

[11] P. Comon. Independent Component Analysis: A new concept. *Signal Processing*, 36:287–314, 1994.

[12] G. Deco and D. Obradovic. *An Information-Theoretic Approach to Neural Computing.* Springer Verlag, 1996.

[13] P. Demartines and J. Hérault. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8:148–154, 1997.

[14] G. H. Golub and C. F. Van Loan. *Matrix Computations, 2nd Ed.* John Hopkins Univ. Press, Baltimore, MD, 1989.

[15] J. Hérault and C. Jutten. Space or time adaptative signal processing by neural network models. In *Neural Networks for Computing. Proceedings of AIP Conference*, pages 206–211, New York, 1986. American Institute of Physics.

[16] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 1997.

[17] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, New York, 1986.

[18] C. Jutten and J. Hérault. Blind separation of sources, part 1: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

[19] J. Karhunen and J Joutsensalo. Representation and separation of signals using nonlinear pca type learning. *Neural Networks*, 7(1):113–127, 1994.

[20] T. Kohonen. *Self-Organizing Maps.* Springer, Berlin, 1995.

[21] M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

[22] R. Linsker. Local synaptic learning rules suffice to maximize mutual information in a linear network. *Neural Computation*, 4(5):691–702, 1992.

[23] D. Lowe and N. Hazarika. Complexity modelling and stability characterisation for long term iterated time series prediction. In *5th IEE International Conference on Artificial Neural Networks*, volume Conference Publication number 440, pages 53–58. The Institute of Electrical Engineers, 1997.

[24] D. J. C. MacKay. Maximum likelihood and covariant algorithms for independent component analysis. in preparation, 1996.

[25] S. Makeig, A. J. Bell, T.-P. Jung, and T.J. Sejnowski. Independent component analysis of electroencephalographic data. In *Advances in Neural Information Processing Systems*, volume 8, pages 145–151. MIT Press, 1996.

[26] J. M. Mendel. Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications. *Proc. IEEE*, 79:278–305, 1991.

[27] C. L. Nikias and J. M. Mendel. Signal processing with higher-order spectra. *IEEE Signal Processing Magazine*, pages 10–37, jul 1993.

[28] B.A. Pearlmutter and L.C. Parra. Maximum likelihood blind source separation: A context-sensitive generalization of ICA. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9 (NIPS*96)*, pages 613–619. MIT Press, Cambridge, MA, 1997.

[29] D. Pham. Blind separation of instantaneous mixture of sources via an independent component analysis. In *IEEE Transactions on Signal Processing*, volume 44, pages 2668–2779, 1996.

[30] H. Pi and C. Peterson. Finding the embedding dimension and variable dependences in time series. *Neural Computation*, 6:509–520, 1994.

[31] J. W. Jr. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18:401–409, 1969.

[32] T. Sauer. Time series prediction using delay coordinate embedding. In A. S. Weigend and N. A. Gershenfeld, editors, *Time Series Prediction: Forecasting the Future and Understanding the Past*, pages 175–193, Reading, MA, 1994. Addison-Wesley.

[33] T. Sauer, J. A. Yorke, and M. Casdagli. Embedology. *Journal of Statistical Physics*, 65:579–616, 1991.

[34] F. Takens. Detecting strange attractors in turbulence. In D. A. Rand and L. S. Young, editors, *Dynamical Systems and Turbulence*, volume 898 of *Lecture Notes in Mathematics*, pages 366–381. Springer, 1981.

Appendix A

A Joint Diagonalization Algorithm

The JADE algorithm uses an extension of the Jacobi technique [14] for diagonalizing a unique hermitian matrix. This technique aims at finding a joint approximate diagonalization of a set $\mathcal{N} = \{N_r | 1 \leq r \leq s\}$ of arbitrary $n \times n$ matrices. It consists in minimizing the diagonalization criterion

$$C(V, \mathcal{N}) = \sum_{r=1}^{s} |\text{diag}(V^H N_r V)|^2 \tag{A.1}$$

by successive Givens rotations. Describing the 2×2 case, let N_r be defined by

$$N_r = \begin{bmatrix} a_r & b_r \\ c_r & d_r \end{bmatrix} \qquad \text{for } r = 1, \ldots, s$$

A complex 2×2 Givens rotation is defined by

$$V = \begin{bmatrix} \cos\theta & -e^{j\phi}\sin\theta \\ e^{j\phi}\sin\theta & \cos\theta \end{bmatrix}$$

Let $a_r^{'}, b_r^{'}, c_r^{'}$ and $d_r^{'}$ denote the coefficients of $V^H N_r V$. Thus, optimization of (A.1) amounts to finding θ and ϕ such that $\sum_r |a_r^{'}|^2 + |d_r^{'}|^2$ is maximized. Noting that $2(|a_r^{'}|^2 + |d_r^{'}|^2) = |a_r^{'} - d_r^{'}|^2 + |a_r^{'} + d_r^{'}|^2$ and that the trace $a_r^{'} + d_r^{'}$ is invariant in a unitary transformation, maximization of criterion (A.1) is equivalent to maximization of Q, defined by

$$Q \stackrel{\text{def}}{=} \sum_r |a_r^{'} - d_r^{'}|^2 \tag{A.2}$$

It can be easily checked that

$$a_r^{'} - d_r^{'} = (a_r - d_r)\cos 2\theta + (b_r + c_r)\sin 2\theta \cos\phi + j(c_r - b_r)\sin 2\theta \sin\phi \qquad (A.3)$$

for $r = 1, \ldots, s$. Then, by defining the vectors

$$\underline{\mathbf{u}} \stackrel{\text{def}}{=} [a_1^{'} - d_1^{'}, \ldots, a_s^{'} - d_s^{'}]^T \qquad (A.4)$$

$$\underline{\mathbf{v}} \stackrel{\text{def}}{=} [\cos 2\theta, \sin 2\theta \cos\phi, \sin 2\theta \sin\phi]^T \qquad (A.5)$$

$$\underline{\mathbf{g}}_r \stackrel{\text{def}}{=} [a_r - d_r, b_r + c_r, j(c_r - b_r)]^T \qquad (A.6)$$

the s equations (A.3) may be written in the form $\underline{\mathbf{u}} = G\underline{\mathbf{v}}$ where $G^T \stackrel{\text{def}}{=} \left[\underline{\mathbf{g}}_1, \ldots, \underline{\mathbf{g}}_s\right]$ so that Q also reads

$$Q = \underline{\mathbf{u}}^H \underline{\mathbf{u}} = \underline{\mathbf{v}}^T G^H G \underline{\mathbf{v}} = \underline{\mathbf{v}}^T Real(G^H G)\underline{\mathbf{v}}$$

where we have used that, $G^H G$ being hermitian by construction, its imaginary part is antisymmetric, hence contributing nothing to the above quadratic form. The last step is to recognize that the particular parameterization (A.5) of $\underline{\mathbf{v}}$ is equivalent to the condition $\underline{\mathbf{v}}^T \underline{\mathbf{v}} = 1$. Thus, the optimal $\underline{\mathbf{v}}$ is the eigenvector of $Re(G^H G)$ associated to the largest eigenvalue, which is easily computed from the coordinates of $\underline{\mathbf{v}}$ without even using trigonometrics as in the standard Jacobi technique [14].

Appendix B

Reconstruction Using Raw Data

In Section 5.2, we have preprocessed the data using the first difference

$$\nabla x(t) = x(t) - x(t-1)$$

instead of the raw data $x(t)$. Here we examine briefly the effect of using the raw data when using the `JADE` algorithm to perform ICA. We are especially interested in the reconstruction of the initial time series using only a few ICs.

If we examine carefully the recovered independent sources (See Figure B.1), we can see that they seem almost identical – which is not particularly good for the reconstruction of the initial time series using only a few ICs.

We can even go further by considering the independent vectors – which are actually the rows of the mixing matrix – in Figure B.2. As for the independent sources, they are almost identical. Such a performance is not surprising because the `JADE` algorithm that is used assumes stationarity of the data provided to it, which unfortunately is not the case with the raw financial data we are concerned with.

From the above, it turns out to be pointless using such transformation to reconstruct the time series.

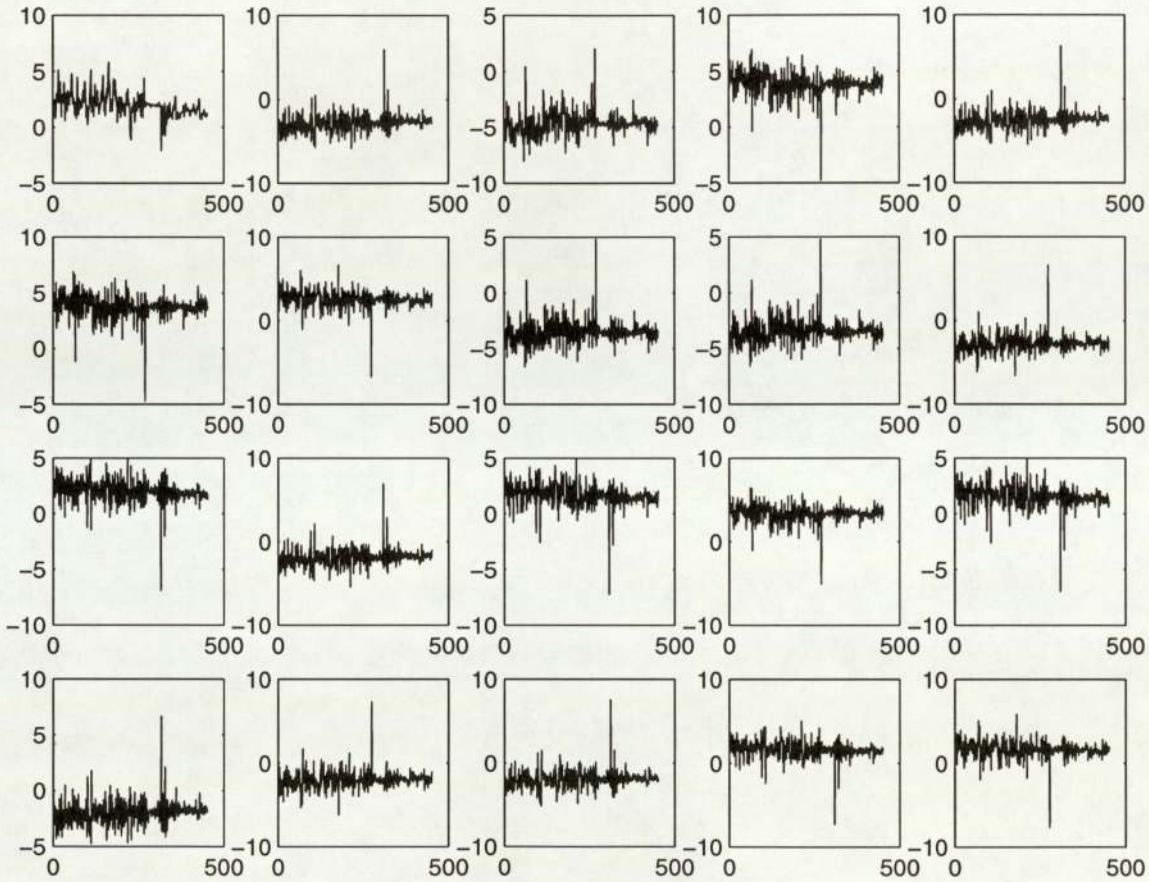


Figure B.1: The first 20 independent components recovered by the JADE algorithm using raw data.

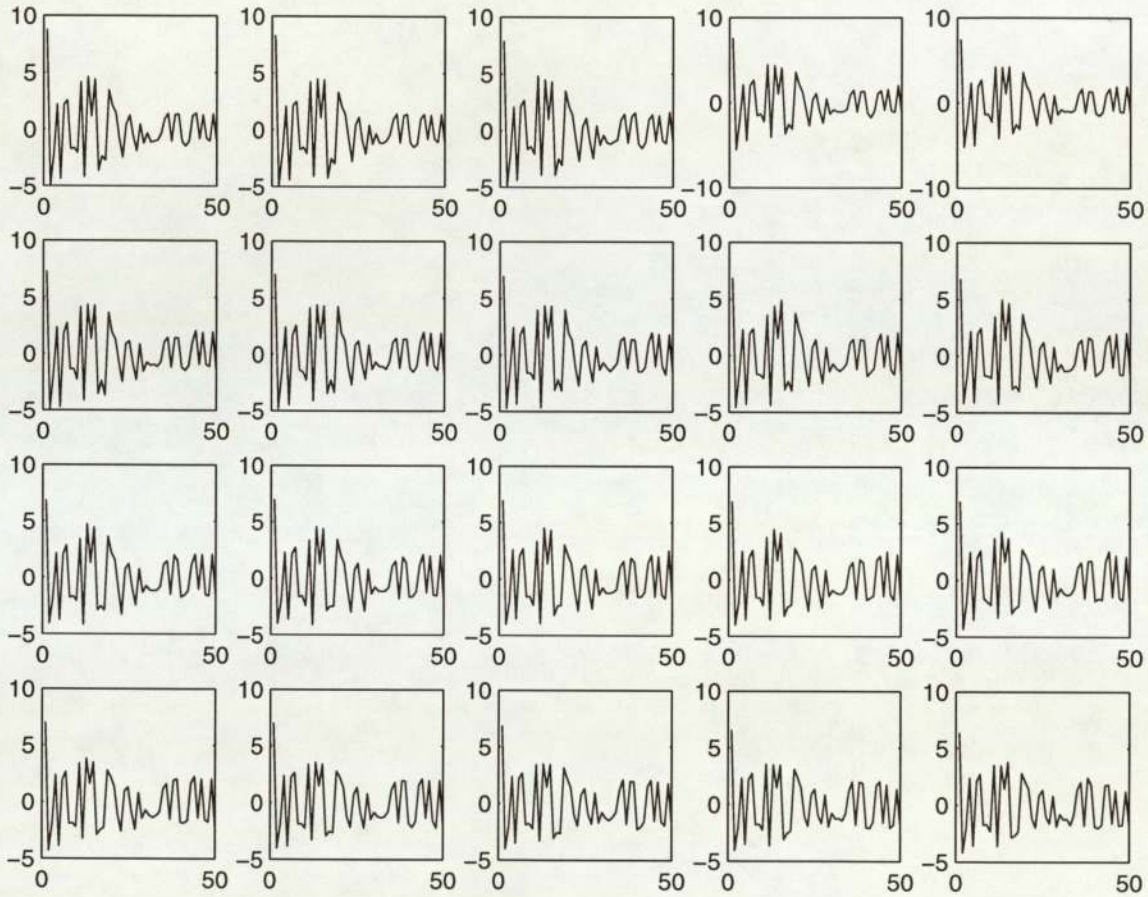


Figure B.2: The first 20 independent vectors recovered by the JADE algorithm using raw data.

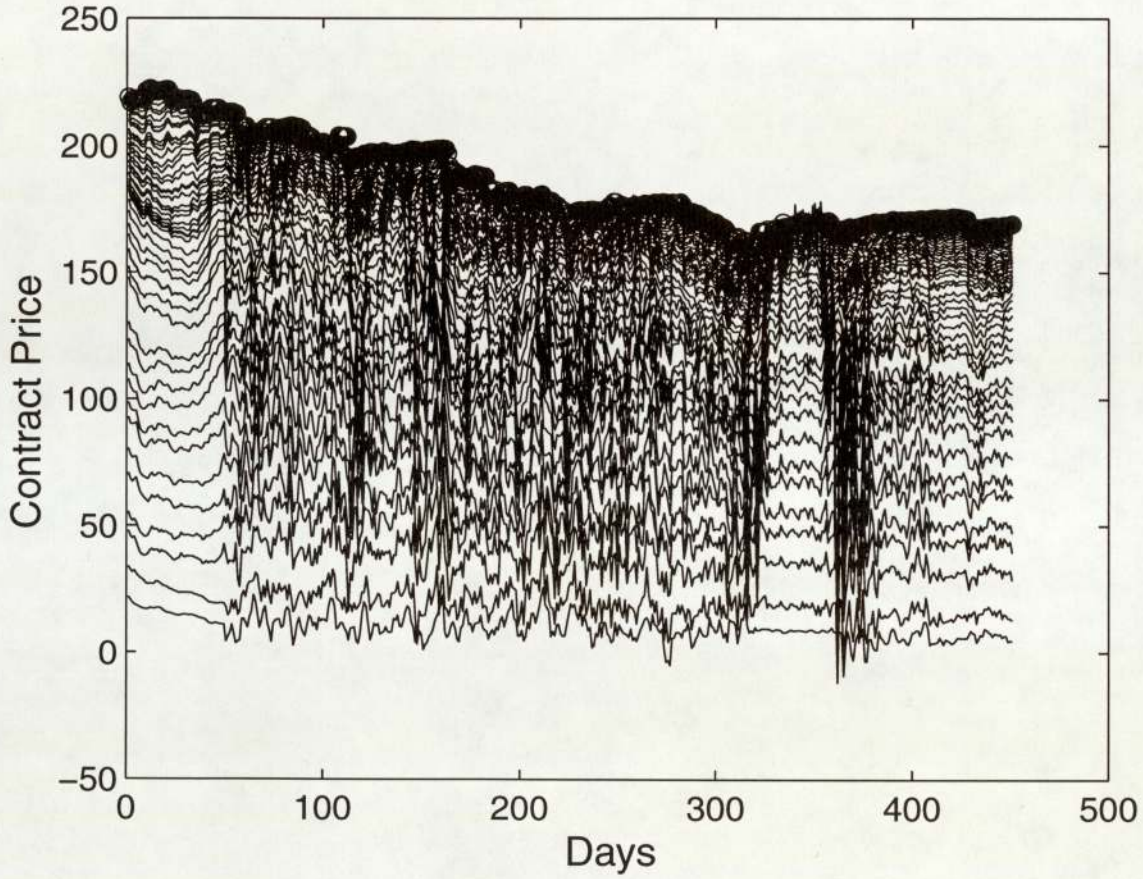


Figure B.3: Reconstruction of the Contract Price time series using the cumulative sum of i ICs, $i = 1, \ldots, 50$. The actual data are indicated by open circles.