Time series forecasting and model selection in singular spectrum analysis

Jacques de Klerk

Dissertation presented for the Degree of Doctor of Philosophy at the University of Stellenbosch

Promoter : Prof. T. de Wet
Date : November 2002

DECLARATION

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

Signature

Date

SUMMARY

Singular spectrum analysis (SSA) originated in the field of Physics. The technique is non-parametric by nature and *inter alia* finds application in atmospheric sciences, signal processing and recently in financial markets. The technique can handle a very broad class of time series that can contain combinations of complex periodicities, polynomial or exponential trend. Forecasting techniques are reviewed in this study, and a new coordinate free joint-horizon k-period-ahead forecasting formulation is derived. The study also considers model selection in SSA, from which it become apparent that forward validation results in more stable model selection.

The roots of SSA are outlined and distributional assumptions of signal series are considered *ab initio*. Pitfalls that arise in the multivariate statistical theory are identified.

Different approaches of recurrent one-period-ahead forecasting are then reviewed. The forecasting approaches are all supplied in algorithmic form to ensure effortless adaptation to computer programs. Theoretical considerations, underlying the forecasting algorithms, are also considered. A new coordinate free joint-horizon k-period-ahead forecasting formulation is derived and also adapted for the multi-channel SSA case.

Different model selection techniques are then considered. The use of scree-diagrams, phase space portraits, percentage variation explained by eigenvectors, cross and forward validation are considered in detail. The non-parametric nature of SSA essentially results in the use of non-parametric model selection techniques.

Finally, the study also considers a commercial software package that is available and compares it with Fortran code, which was developed as part of the study.

OPSOMMING

Singulier spektraalanalise (SSA) het sy oorsprong in die Fisika. Die tegniek is nie-parametries van aard en vind toepassing in velde soos atmosferiese wetenskappe, seinprossesering en onlangs in finansiële markte. Die tegniek kan ‘n wye verskeidenheid tydreekse hanteer wat kombinasies van komplekse periodisiteite, polinomiese- en eksponensiële tendense insluit. Vooruitskattingstegnieke word ook in hierdie studie beskou, en ‘n nuwe koördinaatvrye gesamentlike horison k-periode-vooruitskattingformulering word afgelei. Die studie beskou ook model seleksie in SSA, waaruit duidelik blyk dat voorwaartse validasie meer stabiele model seleksie tot gevolg het.

Die agtergrond van SSA word *ab initio* geskets en verdelingsaannames van seinreekse beskou. Probleemgevalle wat voorkom in die meervoudige statistiese teorie word duidelik geïdentifiseer.

Verskeie tegnieke van herhalende toepassing van een-periode-vooruitskatting word daarna beskou. Die benaderings tot vooruitskatting word in algororitmiese formaat verskaf wat die aanpassing na rekenaarprogrammering vergemaklik. Teoretiese vraagstukke, onderliggend aan die vooruitskattings-algortimes, word ook beskou. ‘n Nuwe koördinaatvrye gesamentlike horison k-periode-vooruitskattingsformulering word afgelei en aangepas vir die multikanaal SSA geval.

Verskillende model seleksie tegnieke is ook beskou. Die gebruik van “scree”-diagramme, fase ruimte diagramme, persentasie variasie verklaar deur eievektore, kruis- en voorwaartse validasie word ook aangespreek. Die nie-parametriese aard van SSA noop die gebruik van nie-parametriese model seleksie tegnieke.

Die studie vergelyk laastens ‘n kommersiële sagtewarepakket met die Fortran bronkode wat as deel van hierdie studie ontwikkel is.

To my parents and family

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to the following persons and institutions:

- My promoter Prof. T. de Wet (Stellenbosch University) and co-promoter Prof. S.J. Steel (Stellenbosch University).
- Prof. J.H. Venter (University of Potchefstroom) for the research topic of this dissertation.
- The National Research Foundation for research funding.
- The Henry Crossley Trust for research funding.
- My parents for their unfailing support and encouragement throughout the years.

SOLI DEO GLORIA

Contents

CHAPTER 1 INTRODUCTION AND NOTATION 1

1.1 Introduction 1

1.2 Thesis contributions 2

1.3 Thesis layout 3

1.4 Notation 4

CHAPTER 2 ELEMENTS OF SSA 10

2.1 Introduction 10

2.2 Roots of SSA 12

2.3 Components of SSA 16

2.3.1 The Trajectory Matrix 16

2.3.2 The Centred Trajectory Matrix 28

2.3.3 The Scatter Matrix 31

2.4 Functions with an Additive Property 32

2.4.1 The Class of Power Functions 33

2.4.2 The Class of Sinusoidal Functions 36

2.4.3 The Class of Polynomials Functions 38

2.4.4 A General Class of Functions 42

2.5 Linear Recurrent Formulae 43

2.6 Noise Models and Statistical Attributes 44

2.6.1 The Additive White Noise Assumption 45

2.6.2 The First-Order Serially Correlated Noise Assumption 49

2.7 Singular Value Decomposition (SVD) in SSA 54

2.7.1 Basics and Definitions 54

2.7.2 The effect of centring in SVD 59

2.7.3 The effect of noise in SVD 60

2.7.4 Optimality of SVD 65

2.7.5 Artificial Phase Portraits 69

2.8 Signal Reconstruction 73

2.8.1 The Hankelization Operation 73

2.8.2 The Approximate Series 75

2.8.3 Signal Reconstruction using Cadzow's Algorithm 76

2.8.4 Other possible signal reconstruction methods 80

2.9 Multi-Channel SSA 81

2.9.1 The Trajectory Matrix 81

2.9.2 The Centred Trajectory Matrix 82

2.9.3 The Scatter Matrix 83

2.9.4 SVD of the Scatter Matrix 83

2.9.5 The Hankelization Operation 83

2.9.6 Reconstruction of an Approximate Series 84

2.9.7 Reconstructing a Signal using Cadzow's Algorithm 85

2.10 Conclusions and Recommendations 86

CHAPTER 3 FORECASTING IN SSA 88

3.1 Introduction 88

3.2 Recurrent one-period-ahead forecasting algorithms for noise-free signals 90

3.2.1 Recurrent one-period-ahead forecasting algorithm due to Danilov (1997) 91

3.2.2 Recurrent one-period-ahead forecasting algorithm due to Venter (1998) 98

3.2.3 An alternative one-period-ahead forecasting algorithm 103

3.2.4 Recurrent one-period-ahead forecasting algorithm due to Golyandina *et al.* (2001) 106

3.2.5 Equivalence of the algorithms of Venter (1998) and Golyandina *et al.* (2001) 109

3.3 Recurrent one-period-ahead forecasting algorithms for noise-contaminated signals 113

3.3.1 Algorithms using an approximate series 115

3.3.2 Algorithms using a reconstructed Cadzow-signal 120

3.4 K-period-ahead forecasting algorithms for noise-free signals 123

3.4.1 Generalized joint-horizon k-period-ahead forecasting algorithm 123

3.4.2 Recurrent vector k-period-ahead forecasting due to Golyandina *et al.* (2001) 127

3.5 K-period-ahead forecasting algorithms for noise-contaminated signals 131

3.5.1 Algorithms using an approximate series 131

3.5.2 Algorithms that use a reconstructed Cadzow-signal 135

3.6 General k-period-ahead forecasting algorithm for multi-channel SSA 135

3.7 Aspects that influence forecasting in SSA 139

3.7.1 Verticality condition 139

3.7.2 Separability 141

3.7.3 End-effect of Hankelization operation 145

3.7.4 Outliers 147

3.7.5 Signal-to-noise ratio 147

3.8 Examples 148

3.8.1 The Airline Passenger Time Series 148

3.8.2 The Hotel Time Series 153

3.8.3 The Monthly Liquor Sales Time Series 156

3.9 Conclusions and Recommendations 160

CHAPTER 4 MODEL SELECTION IN SSA 162

4.1 Introduction 162

4.2 Literature Survey 163

4.3 Choosing the Window Length 165

4.4 Choosing the Number of Eigenvectors 167

4.4.1 Scree-diagrams 167

4.4.2 Cumulative Percentage Variation Explained 171

4.4.3 Artificial Phase Space Portraits and Grouping 172

4.4.4 Cross Validation 176

4.4.5 Forward Validation 186

4.4.6 Robust Forward Validation 220

4.5 Outlier Identification in SSA 224

4.6 Monte Carlo Simulations 229

4.7 Other Examples 238

4.8 Conclusions and Recommendations 245

CHAPTER 5 INFERENCE IN SSA FORECASTING 248

5.1 Introduction 248

5.2 Confidence and Prediction Intervals 249

5.2.1 Confidence intervals for the signal series 249

5.2.2 Prediction intervals for forecasts 252

5.3 Conclusions and Recommendations 265

CHAPTER 6 SSA SOFTWARE 266

6.1 Introduction 266

6.2 CaterpillarSSA Version 3.00 Software Package 266

6.3 SHONGO.F FORTRAN Program 269

6.3.1 Sample output of forward validation model selection 271

6.3.2 Sample output of forecasting 274

6.4 Conclusions and Recommendations 277

CHAPTER 7 SUMMARY AND DIRECTIONS FOR FURTHER RESEARCH 278

APPENDIX 281

Table A-1 International airline passengers: monthly totals 281

Table A-2 Monthly hotel room averages 282

Table A-3 Monthly liquor sales 283

PROGRAM : SHONGO.F 284

REFERENCES 333

Chapter 1

INTRODUCTION AND NOTATION

1.1 Introduction

Singular spectrum analysis (SSA) is a well-known technique in the fields of physics, geophysics and atmospheric sciences. The technique is, however, less known to the statistical community. It is nearly sixteen years old although foundations thereof can be traced back earlier to the field of physics (Takens, 1981). The main paper that laid the foundations for SSA is due to Broomhead and King (1986).

SSA is a non-parametric approach to time series modelling. The technique encapsulates both the strengths of multiple regression analysis and Fourier analysis, albeit in a non-parametric context.

SSA essentially unfolds an observed time series into the column vectors of a Hankel structured matrix, termed the trajectory matrix. In the case of noise-free signals, the column vectors of the trajectory matrix lie on a single r-flat (r-dimensional subspace). Singular value decomposition (SVD) can be used to find the orthonormal base vectors of the linear subspace parallel to this r-flat. This is also where the technique derives its name from, viz. singular spectrum analysis.

Series that are governed by linear recurrent formulae (LRF) can be handled by the technique and, *inter alia,* include combinations of polynomials, exponentials and sinusoidal functions. This insight is due to the groundbreaking article of Buchstaber (1994) and work that followed (Golyandina *et al*., 2001). The technique can therefore be used to model an extremely wide spectrum of time series patterns that mainly involve trend and cyclical behaviour. The strength of the technique lies in its ability to handle multiple cyclical patterns, and also complex cyclical patterns that increase over time.

The unfolding of an observed time series into a trajectory matrix places the SSA methodology into a multivariate framework. In this framework, the statistical distribution theory becomes extremely complex and (mostly) analytically intractable. In this research we were thus inclined towards computer intensive methods. Theoretical results were obtained where possible and numerically intensive approaches were used where this was not possible. The latter was especially true in the case of model selection and inferential procedures that were considered.

Time series forecasting, using SSA, has received attention in the form of recurrent one-period-ahead forecasting (Danilov, 1997; Venter, 1998; Golyandina *et al.*, 2001) and vector forecasting (Golyandina *et al.*, 2001). Other methods of forecasting (Zhang *et al.*, 1993) in SSA exist and predate the advances due to Danilov (1997). These techniques, however, lacked mathematical rigour and are outdated. The major advance in SSA due to Buchstaber (1994) laid foundations for forecasting in SSA. This thesis makes contributions to forecasting and model selection in SSA.

1.2 Thesis contributions

This thesis contains a number of contributions in singular spectrum analysis (SSA). A general formula is derived that can be used to construct joint-horizon k-period-ahead forecasts (see Section 3.4.1 of Chapter 3). Recurrent one-period-ahead forecasting is a special case of the proposed formulation, which is also extended to handle forecasting in the case of multi-channel SSA (see Section 3.6 of Chapter 3). Note that recurrent one-period-ahead forecasting algorithms were available previously (Danilov, 1997; Venter, 1998; Golyandina *et al.*, 2001). A proof is also provided showing the equivalence of the coordinate-free forecasting approach due to Venter (1998) and the recurrent one-period-ahead forecasting approach due to Golyandina *et al.* (2001) (see Section 3.2.5 of Chapter 3).

Model selection is also considered. This has been identified (Venter, 1998; Zhang *et al.*, 1993) as a topic in SSA requiring consideration. This study proposes two approaches towards forward validation model selection in this regard (see Algorithms 4.2 and 4.3 in

Chapter 4). Both approaches utilise the proposed joint-horizon k-period-ahead forecasting algorithm, combined with a bias-variance trade-off measure and a signal reconstruction method not previously used in SSA. This signal reconstruction method is an adaptation of the approach due to Cadzow (1988) and is discussed in Section 2.8.3 of Chapter 2, where further areas of research in this regard are also identified. A small-scale Monte Carlo simulation study and a number of examples are used to show the effectiveness of the proposed methods (see Section 4.6 of Chapter 4).

In Chapter 4, Section 4.5, outlier identification in SSA is considered, using the FAST-MCD program of Rousseeuw and Van Driessen (1999).

Attention is also, briefly, given to inference in SSA. Algorithms based on the bootstrap are proposed for construction of confidence intervals for signal series and prediction intervals for forecasts (see Section 5.2 of Chapter 5).

As a final contribution, FORTRAN software was developed to perform signal reconstruction, cross- and forward validation model selection and forecasting (recurrent and joint-horizon). This software has also been successfully implemented into other languages such as Visual Basic (VB6), Visual Basic for Applications (VBA) and EVIEWS. The FORTRAN code is included as an appendix to the thesis. Due to space restrictions the software for carrying out bootstrap inferences is, however, not included. It is available upon request from the author.

1.3 Thesis layout

Chapter 2 traces back the roots of SSA and outlines the developments in this lively area of research. The technique has recently received a thorough mathematical treatment in Golyandina *et al.* (2001) and opened new possibilities for application in fields such as economics, marketing and finance. Functions that can be handled by SSA are also elaborated upon in Chapter 2.

Chapter 3 is devoted to forecasting in SSA. All the major advances in SSA forecasting are reviewed and algorithmic approaches supplied. A formula is proposed and derived to handle joint-horizon k-period-ahead forecasting. Recurrent forecasting turns out to be a special case of the formulation.

Chapter 4 addresses issues relating to model selection in SSA. Cross validation and forward model validation are considered. A small scale Monte Carlo simulation was also performed as part of the study. Outlier identification and robust model selection procedures are also considered in the Chapter.

Chapter 5, briefly, addresses inferential issues in SSA forecasting. Algorithms that apply the bootstrap are proposed for the construction of confidence intervals for the signal series as well as prediction intervals for forecasts.

Chapter 6 briefly elaborates on new software that perform SSA. Functionality of the CaterpillarSSA version 3.00 (Standard F Edition) software package, as developed by Golyandina *et al.* (2001), is compared with the author's FORTRAN (SHONGO.F) program, which accompanies this thesis as an appendix.

Chapter 7 contains the conclusions and some proposed areas for further research.

1.4 Notation

This section introduces the notation used throughout the thesis.

Vectors and matrices containing sample information are denoted using bold face letters. Bold face Greek letters are reserved for population vectors and matrices. The dimension of a vector and matrix is supplied below such an element, where it is deemed to be necessary. An $n \times p$ matrix will therefore be denoted by,

$$\underset{n\times p}{\mathbf{X}}$$

and an $n \times 1$ vector will be denoted by,

$$\underset{n\times 1}{\mathbf{x}} .$$

The $(i,j)^{th}$ element of a matrix will be denoted by the notation $x_{i,j}$. Tables 1.1 to 1.5, below, can be consulted for notation that will be used throughout this thesis.

Table 1.1 Table of general notation.

GENERAL		
SYMBOL	**DESCRIPTION**	**CHAPTER**
N	Length of an observed time series	2, 3, 4, 5
τ	Window length, dimension of vectors used, number of row vectors in a trajectory matrix	2, 3, 4, 5
n	Number of column vectors in a trajectory matrix such that $n = N - \tau + 1$	2, 3, 4
k	Number of periods forecast ahead	4, 5
$'$	Transpose of a matrix or vector	2, 3, 4
r	Number of orthonormal base vectors that span the parallel linear subspace $\mathfrak{L}_r$	2, 3, 4, 5
$\tilde{\mathbf{x}}$, $\tilde{\mathbf{X}}$	Centred vector or matrix (by subtraction of mean)	2, 3, 4
$\mathbb{R}^{\tau}$	Euclidean space of dimension τ	2, 3, 4
$\mathfrak{L}_r$	Parallel linear subspace spanned by r orthonormal vectors	2, 3, 4
$\mathfrak{L}_r^{\perp}$	Linear subspace orthogonal to $\mathfrak{L}_r$	2, 3, 4

Table 1.2 Table of mathematical operators.

MATHEMATICAL OPERATORS		
SYMBOL	**DESCRIPTION**	**CHAPTER**
$\tilde{\mathbf{X}}$	Centring of matrix, i.e. $\tilde{\mathbf{X}} = \mathbf{X} - \bar{\mathbf{x}}\mathbf{1}'$	2, 3, 4
$\tilde{\mathbf{x}}_i$	Centred column vector, i.e. $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$	2, 3, 4
$\lvert\ \rvert$	Absolute value	2, 3
$\langle \mathbf{x}, \mathbf{y} \rangle$, $\mathbf{x}'\mathbf{y}$	The inner product of two vectors	2, 3
$\angle(\mathbf{x}, \mathbf{y})$	The angle between two vectors	2, 3
$\cos\angle(\mathbf{x}, \mathbf{y})$	The cosine of the angle between two vectors	2, 3
$\lVert\mathbf{x}\rVert = \sqrt{\mathbf{x}'\mathbf{x}}$	Length of a vector ($l_2 - norm$)	2, 3
span($\mathbf{v}_1,\ldots,\mathbf{v}_r$)	Linear space spanned by the vectors $\mathbf{v}_1,\ldots,\mathbf{v}_r$	2, 3
$\mathcal{H}$	Hankelization operator	2, 3
SVD	Singular Value Decomposition	2, 3, 4

Table 1.3 Table of matrix notation.

MATRICES		
SYMBOL	**DESCRIPTION**	**CHAPTER**
$\underset{(\tau\times\tau)}{\boldsymbol{\Sigma}}$	Population covariance matrix	2, 4
$\underset{(\tau\times n)}{\mathbf{X}}$	Trajectory matrix of observed time series (Hankel structured)	2, 3, 4
$\underset{(\tau\times n)}{\tilde{\mathbf{X}}}$	Centred trajectory matrix of observed time series	2, 3, 4
$\underset{(\tau\times n)}{\mathbf{E}}$	Trajectory matrix of noise series (Hankel structured)	2, 3, 4
$\underset{(\tau\times\tau)}{\tilde{\mathbf{X}}\tilde{\mathbf{X}}'}$	Scatter matrix	2, 3, 4
$\underset{(\tau\times r)}{\mathbf{V}}$	Matrix that contains orthonormal base vectors of the linear subspace $\mathfrak{L}_r$ in its columns	2, 3, 4
$\underset{(\tau\times\tau)}{\mathbf{P}_{\mathfrak{L}_r}}$	Projection matrix of the linear subspace $\mathfrak{L}_r$	2, 3, 4
$\underset{(\tau\times\tau)}{\mathbf{P}_{\mathfrak{L}_r^{\perp}}}$	Projection matrix of the linear subspace $\mathfrak{L}_r^{\perp}$	2, 3, 4
$\underset{(\tau\times\tau)}{\mathbf{I}_{\tau}}$	The $\tau\times\tau$ identity matrix	2, 3, 4

Table 1.4 Table of vector notation.

VECTORS		
SYMBOL	**DESCRIPTION**	**CHAPTER**
$\underset{(\tau\times1)}{\boldsymbol{\mu}}$	Population location vector	2, 4
$\underset{(\tau\times1)}{\mathbf{x}_j}$	j^{th} column vector of the trajectory matrix $\mathbf{X}$	2, 3, 4
$\underset{(\tau\times1)}{\tilde{\mathbf{x}}_j}$	j^{th} column vector of the centred trajectory matrix $\tilde{\mathbf{X}}$	2, 3, 4
$\underset{(\tau\times1)}{\overline{\mathbf{x}}}$	Vector of means (based on columns of $\mathbf{X}$), also the shift vector to r-flat H_r	2, 3, 4
$\underset{(n\times1)}{\mathbf{1}}$	Vector of ones, i.e. $\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}'$	2, 3, 4
$\underset{(\tau\times1)}{\mathbf{v}_j}$	j^{th} orthonormal base vector of the linear subspace $\mathfrak{L}_r$	2, 3, 4
$\tilde{\mathbf{x}}_j^{(0,\tau)}$	Vector $\tilde{\mathbf{x}}_j$ with zero placed in the τ^{th} position	2, 3, 4
$\underset{(\tau\times1)}{\mathbf{e}_j}$	j^{th} column vector of the $\tau\times\tau$ identity matrix	2, 3, 4

Table 1.5 Table of time series notation.

TIME SERIES		
SYMBOL	**DESCRIPTION**	**CHAPTER**
$\{f_t^i\}_{t=1}^N$	Deterministic time series (i) of length N	2, 3, 4, 5
$\{\hat{f}_t^i\}_{t=1}^N$	Reconstructed signal series (i) of length N	2, 3, 4, 5
$\{x_t\}_{t=1}^N$	Observed time series of length N	2, 3, 4, 5
$\{e_t\}_{t=1}^N$	Residual series of length N	2, 3, 4, 5
$f(t)$	Continuous signal series at time t	2
$\{e_t^*\}_{t=1}^N$	Bootstrap residual series of length N	5
$\{\hat{f}_t^*\}_{t=1}^N$	Bootstrap reconstructed signal series	5

Chapter 2

ELEMENTS OF SSA

2.1 Introduction

The roots of singular spectrum analysis (SSA) can be traced back to the field of physics (Takens, 1981; Broomhead and King, 1986). Section 2.2 supplies an introduction tracing back the roots of SSA. An historical overview of developments that lead to SSA is given in this section. The section also elaborates on by-products such as the BDS (Brock *et al.*, 1996) statistic that came into existence due to related issues. SSA now finds application in many other fields of which physics, geophysics, atmospheric and climatologic sciences and meteorology are but a few to mention.

Components that make up SSA are thoroughly described, and the mathematical foundations forged, in Section 2.3. Most of the nomenclature of SSA originates from the field of physics, and it is thus imperative for this study to include that section. Statisticians, or any other interested parties for that matter, can therefore familiarise themselves with the terminology and basic theory contained in that section.

The ground-breaking article by Buchstaber (1994) gave mathematical insight and rigour to the underlying mathematical functions that SSA can handle. Section 2.4 is devoted to aspects that resulted from his paper.

The role of linear recurrent formulae (LRF) in the SSA context, was already illustrated in the paper by Buchstaber (1994). A more recent treatment of the issue can be found in Golyandina *et al.* (2001). Section 2.5 briefly takes note of developments in this regard.

Two additive noise models are introduced in Section 2.6. It is important to take note of information contained in this section, as it influences the choice of robust statistical techniques, which is discussed in Chapter 4. It is considered to be an important contribution of this study, as it attempts to illustrate statistical aspects of SSA. The section also underscores the "curse" that the multivariate nature of SSA places on the statistical approach.

Singular value decomposition (SVD) is applied as a dimension reduction technique in SSA. Section 2.7 is devoted to the use of SVD in SSA. SVD is formally introduced and adapted for use in the single-channel SSA scenario.

Section 2.8 discusses signal reconstruction. The section proposes the use of an algorithm, which was proposed by Cadzow (1988) in the field of signal processing. Citation index searches revealed that the Cadzow (1988) algorithm has never been used in SSA, prior to this study.

Section 2.9 briefly introduces multi-channel SSA. The section extends ideas from single-channel SSA to the multivariate time series context. The Cadzow (1988) algorithm is again introduced in a signal reconstruction algorithm, which might be beneficial to the fields of climatologic and atmospheric sciences.

Finally, conclusions are reached and recommendations made in Section 2.10.

2.2 Roots of SSA

The study of dynamic systems is synonymous with the field of physics. Dynamic systems are essentially modelled using differential equations. A convenient notation for a non-autonomous (dependent on time) first-order differential equation is given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t), \tag{2.1}$$

where the single overhead dot denotes the first-order derivative of variable $\mathbf{x}$ with respect to time (t), i.e. $(d\mathbf{x}/dt)$. A state of the system in (2.1) is given by $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots]$.

The movement of a pendulum dangling from a string is, for example, modelled by the second-order differential equation

$$\ddot{x} + \varpi_0^2 \sin(x) = 0, \tag{2.2}$$

where the double overhead dot signifies second-order derivative (d^2x/dt^2). In this representation, x denotes the angle between the displacement of the pendulum's swing and the vertical position of rest. It is therefore used to explain the swinging motion of a pendulum. In the above equation, ϖ_0 denotes the natural frequency of the pendulum. This system is non-linear, but does not display chaotic dynamical behaviour. Chaotic dynamics refers to the phenomenon of a deterministic system that displays a chaotic, i.e. seemingly "random", pattern. At least three differential equations are required in a system to display chaotic behaviour. According to Vautard and Ghil (1989),

"*The classical view, going back to Boltzmann, Gibbs and Einstein, that irregular behaviour in nature results of necessity from the interaction of a large number of degrees of freedom (d-o-f) has been shaken by mathematicians [1], meteorologists [2] and physicists [3]. According to the new dispensation, as few as three d-o-f suffice, by interacting nonlinearly, to create "deterministic chaos".*"

The Lorenz-model is certainly the most well known system of differential equations, and involves three first-order differential equations,

$$\begin{aligned}\dot{x} &= \sigma(y-x)\\ \dot{y} &= rx - y - xz\\ \dot{z} &= -bz + xy.\end{aligned} \tag{2.3}$$

Standard texts (Rasband, 1990) that deal with non-linear dynamics and chaos, usually entertain the Lorenz-model. Differential equations present in dynamic systems, such as the Lorenz-model, are solved using numerical routines. A fourth-order Runga-Kutta routine can be used to solve the Lorenz-model, and the phase space portrait can then be constructed. This would then produce a "time series of the dynamic system".

A system of differential equations, such as the Lorenz-model, is customarily used to construct a so-called phase space portrait. The phase space portrait is a two-dimensional plot of $\dot{x}$ (vertical axis) versus x (horizontal axis). The inventor of this plot was Henri Poincaré (1854-1912), who can be considered as the father of chaos theory. He made valuable contributions to two broad fields of mathematics, namely topology and qualitative theory of differential equations (Bragg, 1998). The phase space diagram is often also called a Poincaré section (Rasband, 1990).

It is clear from Rasband (1990), Broomhead and King (1986) and Gleick (1987) that phase space portraits play a very important role in the analysis of differential equations. According to Gleick (1987),

"*Any state of the system at a moment frozen in time was represented as a point in phase space; all the information about its position or velocity was contained in the coordinates of the point. As the system changed in some way, the point would move to a new position in phase space. As the system changed continuously, the point would trace a trajectory.*"

The study of phase space portraits aids the physicist in investigating chaotic behaviour of a dynamic system. A dynamic system displaying chaotic behaviour has a so-called strange attractor, which can be identified by using a phase space portrait. Hence, the presence of chaos can be identified using such a diagram.

The phase space portrait can alternatively be constructed using a method called *time delays*. This method is due to Packard *et al.* (1980) and Takens (1981). It essentially entails the fact that one usually has an experimental set of data, rather than the actual set of linear differential equations governing a dynamical system. The problem is therefore the reverse of a theoretical setting. Faced with an experimental set of discretely sampled time series data, what would the underlying set of linear differential equations be? If one could not obtain these differential equations, could one at least divulge something of the system's behaviour through study of phase space portraits? This problem setting paved the way towards the idea of *time delays*. According to Elsner and Tsonis (1996, pp. 40-41)

"It is assumed that the variables in the evolution of the dynamical system satisfy a set of p *first-order differential equations, that is,*

$$\begin{aligned}\dot{x}_1 &= f_1\left(x_1, x_2, \dots, x_p\right)\\ \dot{x}_2 &= f_2\left(x_1, x_2, \dots, x_p\right)\\ &\vdots\\ \dot{x}_p &= f_p\left(x_1, x_2, \dots, x_p\right)\end{aligned}$$

... Ruelle (1980) suggested that instead of a continuous variable and its derivative, a discrete time series x_t *and its successive shifts by a lag parameter* τ *should be enough to describe the dynamics of the system. The shifting amounts to a first-order differencing of the discrete time series. The first-order differencing of a time series is the analogous of differentiation. Therefore, given an observed time series, one can consider lagged copies of the series as additional variables of the system.*

...by using lagged copies of a single time series we can define the coordinates of the phase space that will approximate the dynamics of the system from which the time record was sampled. The number of lags is called the embedding dimension. Because the above

method uses lagged (or delayed) copies of segments of a time series, the procedure is referred to as the method of delays.

For the purpose of SSA the method of delays is the procedure that takes a univariate time record and makes it a multivariate set of observations."

The paper by Broomhead and King (1986) followed the work done by Packard *et al.* (1980) and Takens (1981). In their paper they introduced the notion of a *trajectory matrix*. This matrix contains copies of the delayed time series segments in its column vectors. The trajectory matrix would then have the following format

$$\begin{aligned}\mathbf{X} &= \begin{bmatrix} f_1 & f_2 & \cdots & f_n \\ f_2 & f_3 & \cdots & f_{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ f_\tau & f_{\tau+1} & \cdots & f_N \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix}. \end{aligned} \tag{2.4}$$

According to Broomhead and King (1986) the entire observed time series is unfolded into the column vectors of the trajectory matrix displayed in (2.4). The column vectors in this matrix are therefore embedded in $\mathbb{R}^\tau$ where τ is known as the embedding dimension, i.e $\{\mathbf{x}_i \in \mathbb{R}^\tau, i = 1,\dots,n\}$. This concept was termed the "$\tau -$window". The main contribution of their paper then concentrated on singular value decomposition (SVD) of the trajectory matrix. More attention will be given to this in ensuing sections.

Finally, an interesting by-product of the methodology discussed so far, is a test statistic called the BDS statistic (Brock *et al.* 1996; LeBaron, 1997). Statisticians were obviously intrigued by the notion of a set of differential equations that exhibited chaotic outcomes. This type of outcome, when viewed in a time series format, seemed random. The general question was then posed as to whether random was truly random or whether a seemingly random structure could be described by a dynamic system with chaotic outcome. If the latter was the case, then predictability was a distant possibility. A way then had to be found to predict such a time series. This thesis will elaborate upon this notion in due course and make contributions in the forecasting paradigm.

The BDS statistic already features in reputable software packages and seems to be widely used. According to the EVIEWS 4 (Quantitative Micro Software, 2000) software package manual,

"*The BDS test is a portmanteau test for time based dependence in a series. It can be used for testing against a variety of possible deviations from independence including linear dependence, non-linear, or chaos.*"

The study of chaotic behaviour of differential equations did not confine itself to the field of physics. It was not long before "Chaos" became standard nomenclature in fields that had phenomena with similar "seemingly random" behaviour. Even professionals in capital markets (Peters, 1991) joined the debate. It would seem that many fields had some form of time series that exhibited random pattern. Since there was now a way of distinguishing between random and chaos, unpredictability and predictability, many articles were published under the "Chaos" banner.

This section was intended to shed some light on the roots that SSA can be traced back to. The essential components of SSA will be considered now and the underlying mathematical foundations be forged.

2.3 Components of SSA

2.3.1 The Trajectory Matrix

The first paper to deal with the notion of a trajectory matrix was published by Broomhead and King (1986). The idea followed work done by Packard *et al.* (1980) and Takens (1981), who introduced the method of time delays in studying qualitative properties of dynamic systems. The definition of a trajectory matrix is formalised below.

Definition 2.1 Trajectory Matrix

A *Trajectory Matrix* based on an observed time series $\{f_t\}_{t=1}^{N}$ of length N, is a $\tau \times n$ Hankel structured matrix $\mathbf{X}$ formed using the rule

$$x_{ij} = f_{i+j-1} \tag{2.5}$$

for $i = 1,\dots,\tau$ and $j = 1,\dots,n(= N - \tau + 1)$. Thus

$$\begin{aligned} \underset{(\tau \times n)}{\mathbf{X}} &= \begin{bmatrix} f_1 & f_2 & f_3 & \cdots & f_n \\ f_2 & f_3 & f_4 & \cdots & f_{n+1} \\ f_3 & f_4 & f_5 & \cdots & f_{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_\tau & f_{\tau+1} & f_{\tau+2} & \cdots & f_N \end{bmatrix} \\ &\equiv \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_n \end{bmatrix}. \end{aligned} \tag{2.6}$$

It is clear from the definition of a *trajectory matrix* that the entire observed time series is *unfolded* into the column vectors of this matrix. The dimension of the column vectors (τ) will be referred to as the *window length*. The window length is clearly the dimension of the Euclidean space into which the time series is unfolded, and the choice thereof is restricted to

$$2 \le \tau \le \text{integer part of } (N+1)/2. \tag{2.7}$$

The number of column vectors (n) in the trajectory matrix is dependent on the number of observations (N) in an observed time series, and is restricted by

$$N - (\text{integer part of } (N+1)/2\ ) + 1 \le n \le (N-1). \tag{2.8}$$

The trajectory matrix has a very special structure. From perusal of (2.6) it should be evident that the reverse diagonal elements of this matrix are all the same. This is an extremely important feature of the trajectory matrix, as this is the reason behind the success of SSA. The type of structure matrix in (2.6) is termed a Hankel structured matrix. In statistical terms the number of column vectors (n) in the trajectory matrix represents the sample size, in the current context. The number of row vectors (τ) represents the number of variables, in the current context. It is clear that the trajectory matrix places a univariate time series into a multivariate framework.

In general multivariate statistical texts (Flury, 1997; Morrison, 1990) the reverse to the above usually holds. The number of row vectors in a matrix usually denotes the sample size, typically the number of entities studied. The number of column vectors usually represents the number of variables, and is considered the number of measurements taken per entity. It is important to note this slight deviation in notation before proceeding. It is also important to recall that the row vectors of a typical matrix of multivariate observations, say the $n \times p$ matrix $\mathbf{Y}$, is usually assumed to be multivariate normally distributed, i.e. that

$$\mathbf{y}_i \sim N_p(\boldsymbol{\mu}, \ \boldsymbol{\Sigma}) \tag{2.9}$$

for $i = 1,...,n$. Attention will be given to the multivariate distribution of trajectory matrix column vectors in a later section, where assumptions regarding noise distributions are coupled to the SSA methodology.

Buchstaber (1994) noted that if a deterministic time series is unfolded into the column vectors of a trajectory matrix, that all the column vectors of the trajectory matrix lie on a single r-flat (H_r). He also proposed a general class of continuous functions, for which this is true. This general class of functions will be considered in greater detail in due course.

Turning back to the notion of an r-flat, Venter (1998) illustrated how the r-flat H_r could be considered as a combination of a shift vector $(\mathbf{b})$ and parallel linear subspace $\mathfrak{L}_r$, where $\mathfrak{L}_r = span(\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_r)$. The vectors $\mathbf{v}_1, ..., \mathbf{v}_r$ represent a base for the parallel linear subspace $\mathfrak{L}_r$.

Definition 2.2 r**-flat**

An r*-flat* ($H_r \subset \mathbb{R}^\tau$) with shift vector $\mathbf{b}$ and parallel linear subspace $\mathfrak{L}_r$, spanned by the base vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_r$, is defined as

$$H_r = \{\ \mathbf{x} : \mathbf{x} = \mathbf{b} + a_1\mathbf{v}_1 + ... + a_r\mathbf{v}_r;\ a_i \in \mathbb{R};\ \mathbf{v}_i \in \mathbb{R}^\tau; \tau \geq r+1\} \qquad (2.10)$$

The vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_r$ form a base for the parallel linear subspace and are therefore linearly independent. It is also clear that a 1-flat in $\mathbb{R}^\tau$ is a line, that a 2-flat in $\mathbb{R}^\tau$ is a plane and that a $(\tau - 1)$-flat is a hyperplane (see Fraleigh and Beauregard, 1987, p. 296).

In this study the base vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_r$ will be obtained through singular value decomposition (SVD). SVD yields mutually orthogonal and normalized vectors, such that

$$\mathbf{v}_i'\mathbf{v}_j = 0 \textit{ if } i \neq j$$
$$\mathbf{v}_i'\mathbf{v}_j = 1 \textit{ if } i = j$$

for $i = 1, ..., r$ and $j = 1, ..., r$.

The fact that these vectors are mutually orthogonal, guarantees that they are linearly independent, and therefore form a base for the parallel linear subspace.

The shift vector ($\mathbf{b}$) of r-flate H_r is unique and needs to be calculated only once. When r-flat H_r is spanned by r base vectors, it is said to be of rank r. In such a case the minimum window length is set at $\tau \geq r+1$.

The trajectory matrix in (2.6) has a Hankel structure because of the way in which an observed time series is unfolded into the column vectors. A Hankel structured matrix has identical reverse diagonal elements. It is also possible to change the unfolding rule and obtain a Toeplitz structured matrix. This approach seems to be more widely used in the field of signal processing (Cadzow, 1988). The Toeplitz form of the trajectory matrix is given in (2.11),

$$\underset{\tau\times n}{\mathbf{X}} = \begin{bmatrix} f_{N-\tau+1} & f_{N-\tau+2} & f_{N-\tau+3} & \cdots & f_1 \\ f_{N-\tau+2} & f_{N-\tau+3} & f_{N-\tau+4} & \cdots & f_2 \\ f_{N-\tau+3} & f_{N-\tau+4} & f_{N-\tau+5} & \cdots & f_3 \\ \cdots & \cdots & \cdots & \vdots & \vdots \\ f_N & f_{N-1} & f_{N-2} & \cdots & f_\tau \end{bmatrix}. \tag{2.11}$$

A few examples will now be used to illustrate the characteristics of the column vectors of the trajectory matrix.

Example 2.1

In this example a series was generated as

$$x_t = f_t + \varepsilon_t \ , \ t = 1,\ldots,100, \tag{2.12}$$

where

$$\begin{aligned} f_t &= 10 + 2t \\ \varepsilon_t &\sim N\left(0,\ 10^2\right). \end{aligned} \tag{2.13}$$

The series was unfolded into a trajectory matrix and the column vectors plotted (cf. Figure 2.1 below). The window length was set at $\tau = 2$ to visualize the example in two-dimensional Euclidean space. The first pane in Figure 2.1 displays the white noise contaminated series $\{x_t\}_{t=1}^{N}$ and also the noise-free series $\{f_t\}_{t=1}^{N}$. The second pane illustrates the two-dimensional plot of the column vectors of the trajectory matrix, formed using both the white noise contaminated and noise-free series.

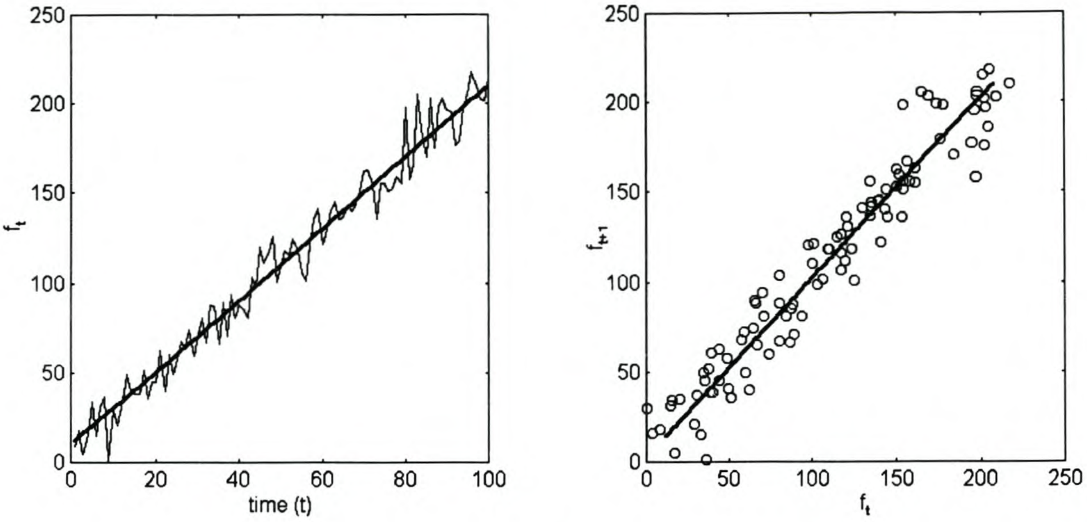


Figure 2.1 Plot of column vectors of trajectory matrix in $\mathbb{R}^2$ ***(first-order polynomial series).***

Circles are used to display the column vectors of the white noise contaminated series. The noise-free series was also unfolded into a trajectory matrix. The column vectors of the series all fall on a single "plane" in the latter case, i.e. the line in this figure.

It is evident from this illustration that noise causes the column vectors of a trajectory matrix not to lie on a single plane. This already hints towards the use of a method to reconstruct a plane as close as possible to all the observed points.

Example 2.2

This example illustrates the unfolding of a series generated from a sinusoidal function and white noise, viz.

$$x_t = f_t + \varepsilon_t \ , \ t = 1,...,100,$$

where

$$\begin{aligned} f_t &= 150\sin(2\pi t / 12 + 100) \\ \varepsilon_t &\sim N\left(0,\ 100^2\right). \end{aligned} \tag{2.14}$$

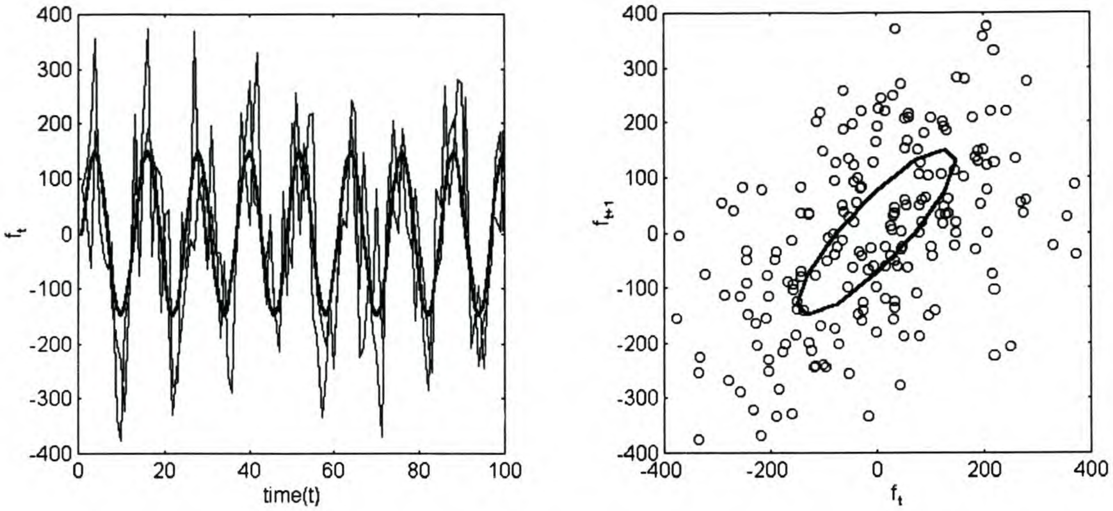


Figure 2.2 Plot of column vectors of trajectory matrix in $\mathbb{R}^2$ (sinusoidal series).

The noise-free and white noise contaminated series are given in the first panel of Figure 2.2. The second panel again contains the plot of trajectory matrix column vectors. In this case the window length was set at $\tau = 2$. The column vectors of the trajectory matrix were therefore unfolded in two-dimensional Euclidean space $\left(\mathbb{R}^2\right)$.

It is clear from the second panel that a strange figure represents the trajectory matrix, formed using the noise-free series. This shape has 12 sides and cannot be considered as a plane in $\mathbb{R}^2$. On closer inspection of the function that was simulated in (2.14), it is clear that the period of this sinusoid is 12. This is also the number of sides of the shape in Figure 2.2. This phenomenon will be investigated in examples that follow.

The window length was then set equal to $\tau = 3$ and resulted in Figure 2.3 below. From this representation it follows that the noise-free series has column vectors lying on a single plane in $\mathbb{R}^3$. This fact will become clear in Section 2.4.2, when the base vectors for the plane, on which the column vectors of the trajectory matrix lie, are derived.

It is an interesting notion of increasing the window length (τ) until it becomes apparent that the multivariate scatter of data "closely" surrounds an r-flat (or possibly any other multivariate structure). Unfortunately this procedure can only be applied up to a window length $\tau = 3$, since it is the highest dimension that can be physically graphed. If the multivariate scatter of data dissipates even more as the window length increases, the series under consideration is most probably a white noise series. White noise is like a gas and will explore any dimension it is placed in, in a uniform manner. The higher the dimension, i.e. larger window length (τ), the more a white noise series will explore the dimension. It will not prefer to lump close to any r-flat (or any other multivariate structure). This is a way of distinguishing between predictability and unpredictability of a series of observations in the context of SSA.

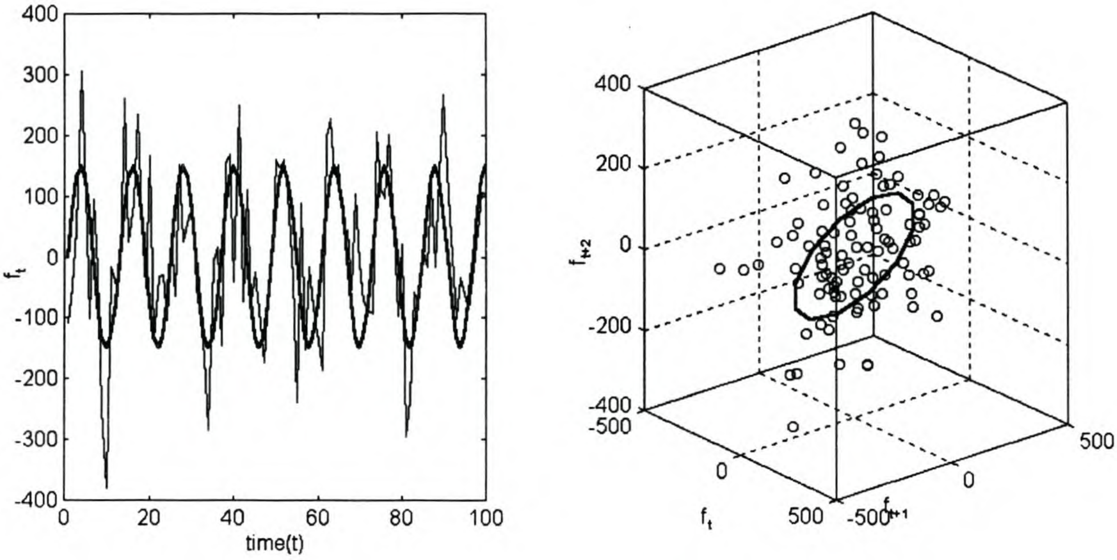


Figure 2.3 Plot of column vectors of trajectory matrix in $\mathbb{R}^3$ ***(sinusoidal series with period 12).***

Example 2.3

It was found, in the previous example, that the period of a sinusoidal gave rise to a plane in $\mathbb{R}^3$, with number of sides equalling the period. To illustrate this phenomenon for another period, a series from the following model was generated

$$x_t = f_t + \varepsilon_t \ , \ t = 1,\ldots,100,$$

where

$$\begin{aligned} f_t &= 150\sin(2\pi t/6 + 100) \\ \varepsilon_t &\sim N\left(0,\ 100^2\right). \end{aligned} \tag{2.15}$$

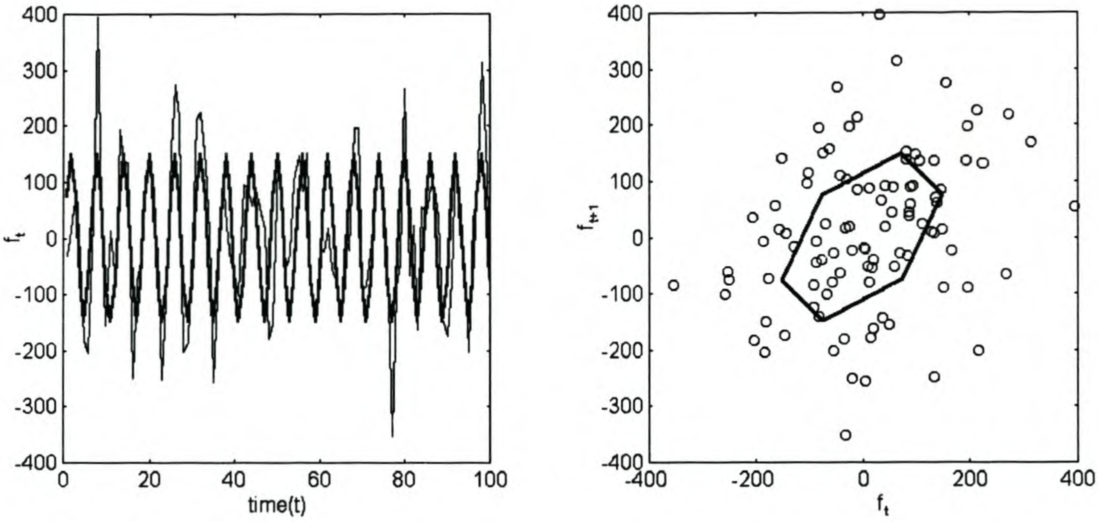


Figure 2.4 Plot of column vectors of trajectory matrix in $\mathbb{R}^2$ ***(sinusoidal series with period 6).***

The period of the simulated sinusoid is 6. This is coincidentally also the number of sides of the plane in $\mathbb{R}^3$, cf. Figure 2.5 below.

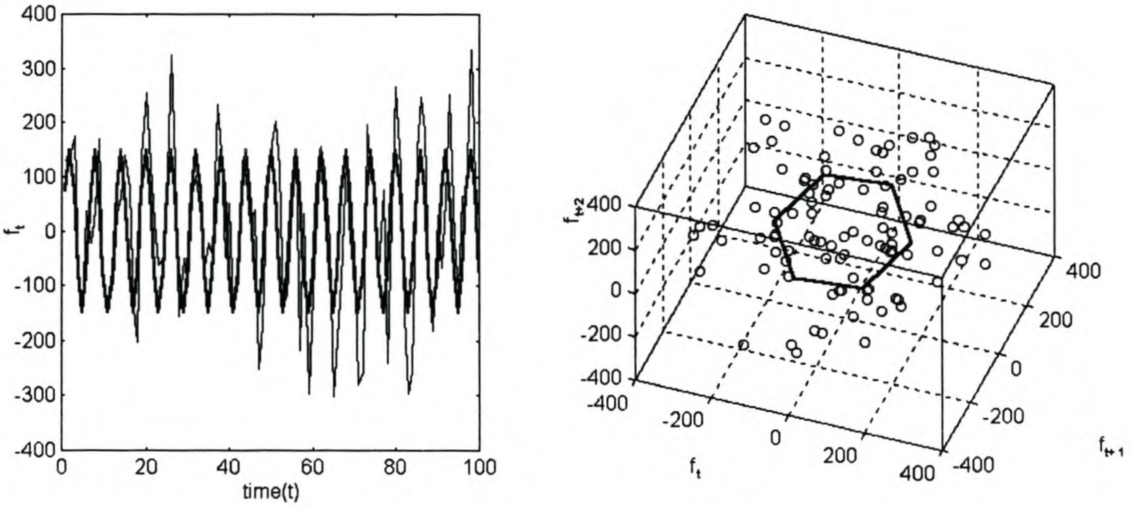


Figure 2.5 Plot of column vectors of trajectory matrix in $\mathbb{R}^3$ ***(sinusoidal series with period 6).***

Example 2.4

As a last example on the effect of a sinusoid's period, a series was generated from

$$x_t = f_t + \varepsilon_t \ , \ t = 1,\ldots,100,$$

where

$$\begin{aligned} f_t &= 150\sin(2\pi t / 4 + 100) \\ \varepsilon_t &\sim N\left(0,\ 100^2\right). \end{aligned} \tag{2.16}$$

It is again clear from Figure 2.6 and Figure 2.7 that the period of a sinusoid gives rise to an interesting plane in $\mathbb{R}^3$. It is also clear that the plane on which the column vectors of a sinusoid's trajectory matrix fall, is spanned by two vectors. These properties will receive thorough mathematical attention in §2.6 below.

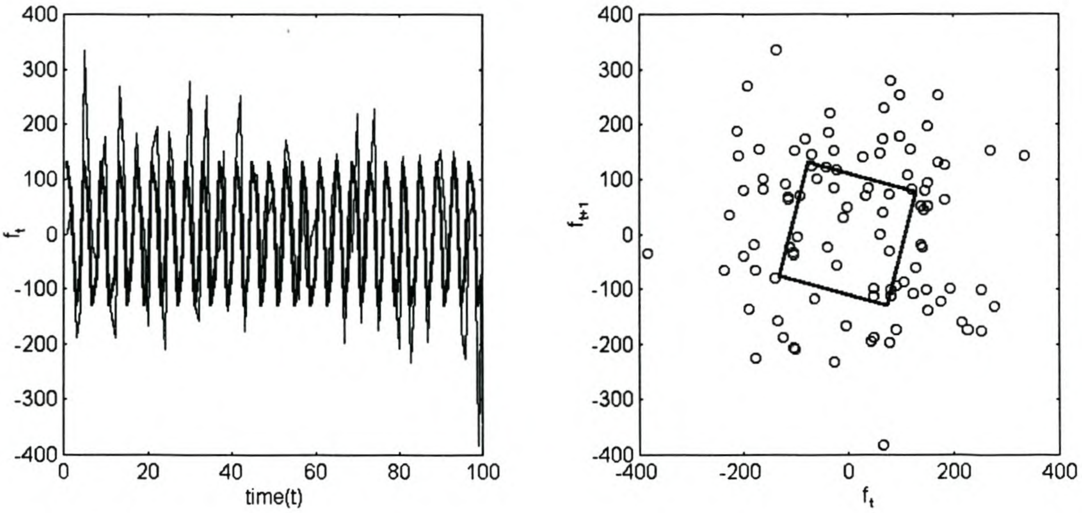


Figure 2.6 Plot of column vectors of trajectory matrix in $\mathbb{R}^2$(sinusoidal series with period 4).

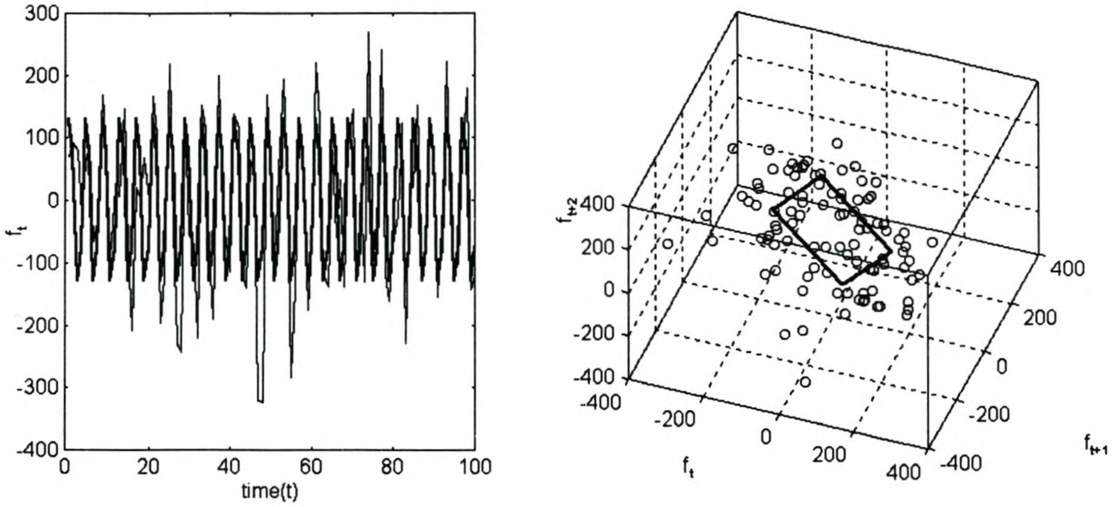


Figure 2.7 Plot of column vectors of trajectory matrix in $\mathbb{R}^3$ ***(sinusoidal series with period 4).***

Example 2.5

In the final example of this section, a series was generated from

$$x_t = f_t + \varepsilon_t \ , \ t = 1,...,100,$$

where

$$\begin{aligned} f_t &= 500 + t + 0.1t^2 \\ \varepsilon_t &\sim N\left(0,\ 500^2\right). \end{aligned} \tag{2.17}$$

Figure 2.8 below displays the resulting simulated series.

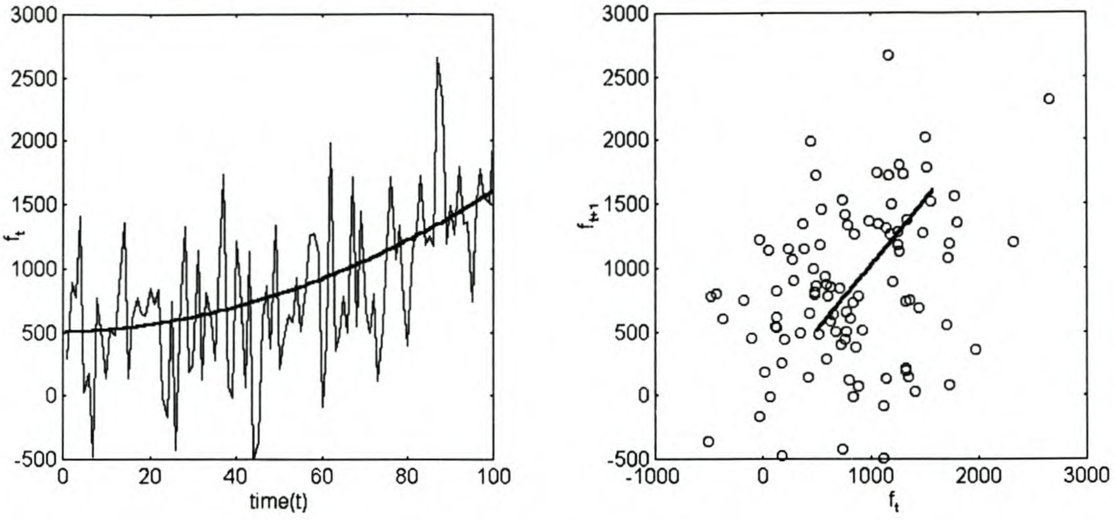


Figure 2.8 Plot of column vectors of trajectory matrix in $\mathbb{R}^2$ (second-degree polynomial).

The left panel in Figure 2.8 above exhibits the noise-free and noise-contaminated signals. The panel to its right contains the scatter of the trajectory matrix column vectors. It is not too clear if a line in $\mathbb{R}^2$ or a plane in $\mathbb{R}^3$ best approximates the column vectors of the noise-free signal. This issue will be studied in §2.4.3 below.

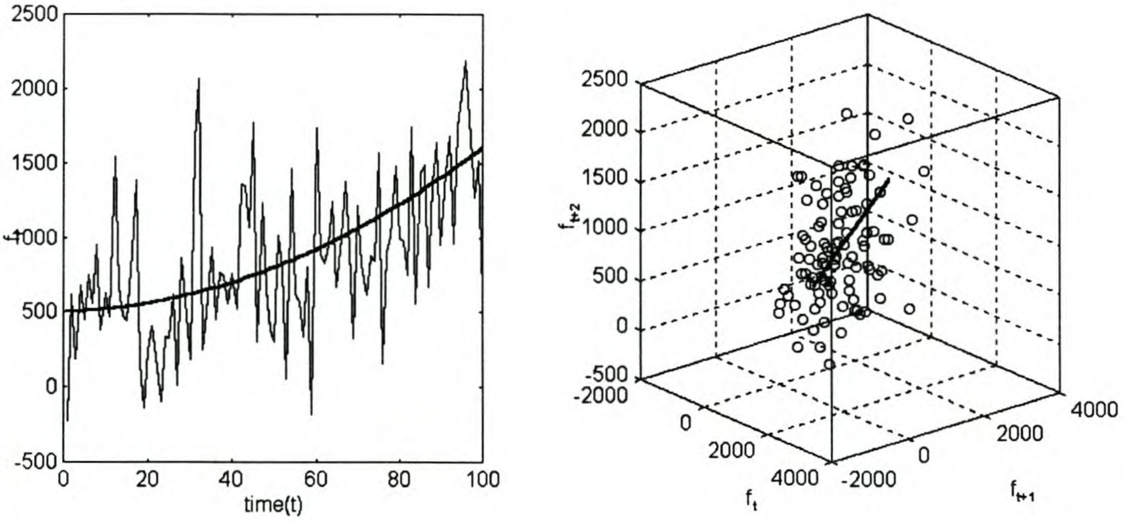


Figure 2.9 Plot of column vectors of trajectory matrix in $\mathbb{R}^3$ (second-degree polynomial).

We now shift attention to the centred trajectory matrix in the next paragraph.

2.3.2 The Centred Trajectory Matrix

It is customary in the multivariate statistical literature to centre a matrix of multivariate observations, by subtracting the vector of means from each row vector, when calculating the sample covariance matrix. The vector of means is calculated by summation over the row vectors, which constitute the sample of observations. The vector of means has dimension equalling the number of columns in the matrix of observations. This vector is subtracted from each row vector, thereby centring each of the row vectors. Geometrically, this operation results in a centring of the multivariate scatter of data at the origin.

In SSA, the number of columns represents sample size, and the number of row vectors represents the number of variables. The vector of means is calculated over the column vectors. Each column vector is then centred about the origin, by subtracting the vector of means from each of them. In geometrical terms, the cloud of multivariate points is then centred about the origin. No information is lost in this process. The variance and covariance that exist between the variables, remain unchanged. This is a very important feature of centring, and does not influence any analysis that is based on the centred matrix. The centred trajectory matrix will now formally be defined.

Definition 2.3 Centred Trajectory Matrix

A *Centred Trajectory Matrix* is a $\tau \times n$ matrix defined by

$$\underset{(\tau \times n)}{\tilde{\mathbf{X}}} = \mathbf{X} - \bar{\mathbf{x}}\mathbf{1}', \tag{2.18}$$

where

$$\underset{(\tau \times 1)}{\bar{\mathbf{x}}} = \frac{1}{n}\mathbf{X}\mathbf{1}. \tag{2.19}$$

The vector of means is denoted by $\bar{\mathbf{x}}$ in equation (2.18) and vector $\mathbf{1}'$ is simply the transpose of the unit vector, with dimension equal to the number of column vectors in matrix $\mathbf{X}$. The vector of means $(\bar{\mathbf{x}})$ will be used as the shift vector of r-flat H_r. Recall that all the column vectors of a trajectory matrix fall on r-flat H_r, in the case of a time series originating from the broad class proposed by Buchstaber (1994).

The r-flat H_r will therefore, from now on, be written in the form

$$H = \{\ \mathbf{x} : \mathbf{x} = \overline{\mathbf{x}} + a_1\mathbf{v}_1 + \ldots + a_r\mathbf{v}_r;\ a_i \in \mathbb{R};\ \mathbf{v}_i \in \mathbb{R}^{\tau}; \tau \geq r+1\}, \qquad (2.20)$$

where vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r$ form a base for parallel linear subspace $\mathfrak{L}_r$ and the vector of means is defined as in (2.19).

Geometrically, r-flat H_r will have the form,

$$\underbrace{H_r}_{r\text{-flat}} = \underbrace{\overline{\mathbf{x}}}_{\text{shift vector}} + \underbrace{\mathfrak{L}_r}_{\substack{\text{parallel}\\ \text{linear subspace}}} . \qquad (2.21)$$

The examples considered in §2.3.1 will now be re-considered using the centring technique.

Example 2.6

Figure 2.10 displays the effect of centring the trajectory matrix column vectors. The two elliptical structures highlight the previous and current position of column vectors.

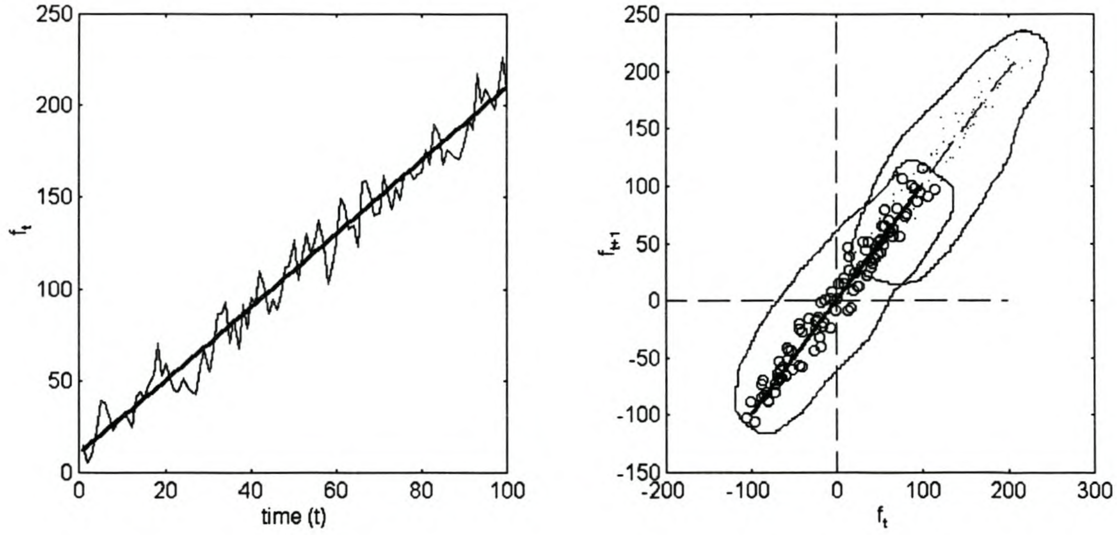


Figure 2.10 Plot of Column Vectors of Centred Trajectory Matrix in $\mathbb{R}^2$**.**

Example 2.7

The column vectors, formed from the simulated sinusoid series, were centred in this example.

It is evident from the right panel in Figure 2.11 below, that the scatter of points is centred about the origin. The plane, on which the column vectors of the noise-free series reside, has also centred itself about the origin.

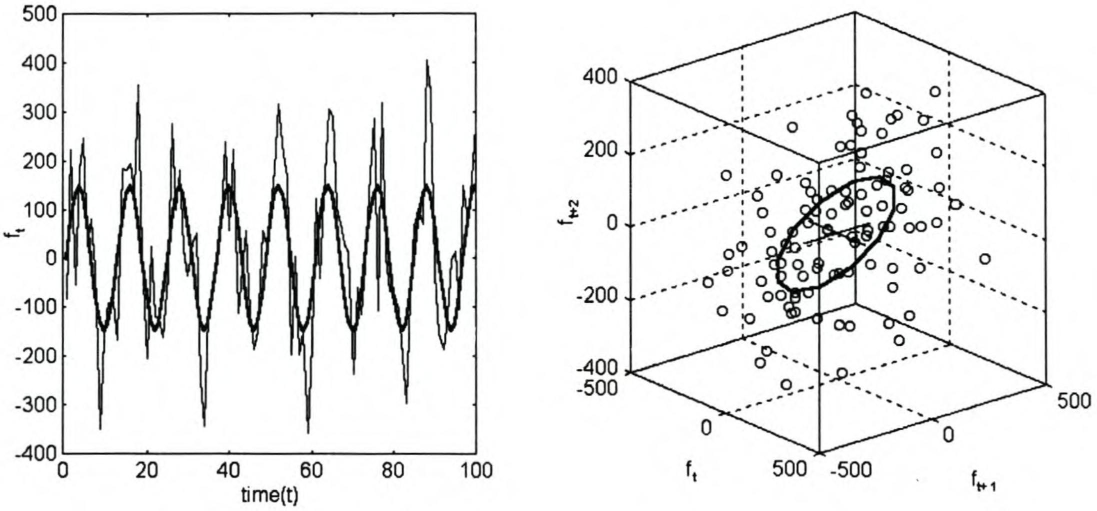


Figure 2.11 Plot of Column Vectors of Centred Trajectory Matrix in $\mathbb{R}^3$.

It is not necessary to repeat the centring procedure for all the sinusoid examples. The idea remains the same.

Example 2.8

It is clear from Figure 2.12, below, that centring also had the same geometrical effect of a shift and centring of scatter about the origin.

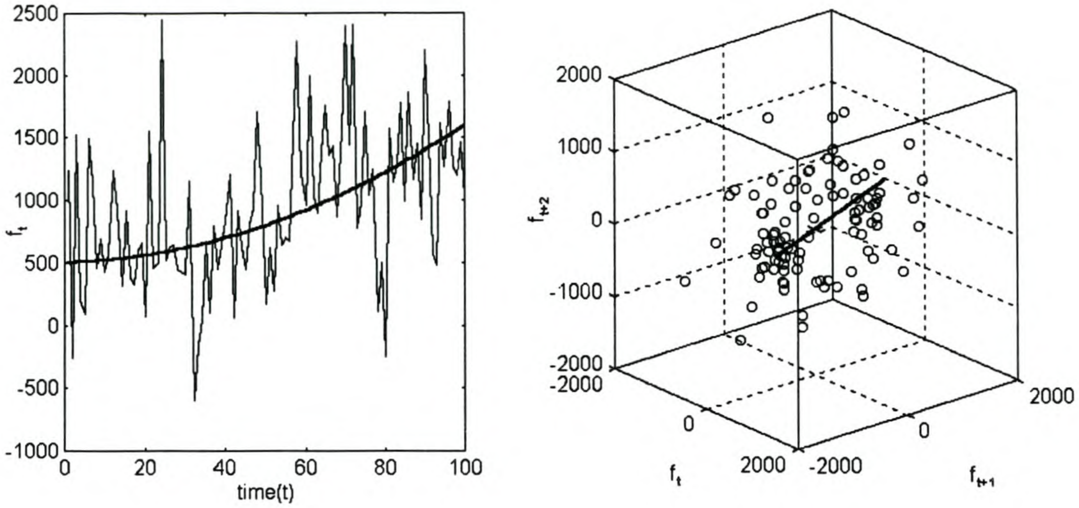


Figure 2.12 Plot of Column Vectors of Centred Trajectory Matrix in $\mathbb{R}^3$.

2.3.3 The Scatter Matrix

Definition 2.4 Scatter Matrix

A *Scatter Matrix* is a symmetric $\tau \times \tau$ matrix defined by either,

$$\underset{(\tau\times\tau)}{\tilde{\mathbf{X}}\tilde{\mathbf{X}}'} = (\mathbf{X}-\bar{\mathbf{x}}\mathbf{1}')(\mathbf{X}-\bar{\mathbf{x}}\mathbf{1}')' \tag{2.22}$$

or

$$\underset{(\tau\times\tau)}{\mathbf{X}\mathbf{X}'}. \tag{2.23}$$

The scatter matrix will be used in singular value decomposition (SVD). If the scatter matrix is elementwise divided by $(n-1)$ then it is the usual multivariate sample covariance matrix $\mathbf{S} = \frac{1}{n-1}(\mathbf{X}-\bar{\mathbf{x}}\mathbf{1}')(\mathbf{X}-\bar{\mathbf{x}}\mathbf{1}')'$. Subtracting the vector of means from a matrix of multivariate observations is required to calculate the covariance matrix $\mathbf{S}$.

2.4 Functions with an Additive Property

In the article by Buchstaber (1994), he illustrated how the $(i,j)^{th}$ element of a trajectory matrix can be linked to functions that have an "additive property". In his article he showed how the trajectory matrix, formed from a time series $\{f_t\}_{t=1}^{N}$, discretely sampled from a continuous function $f(t)$ with an additive property, has column vectors lying on a single r-flat H_r.

A function that has an additive property is defined in Definition 2.5 below.

Definition 2.5 Functions with an Additive Property

A continuous function $f(t)$ has an additive property if it can be written as a combination of continuous functions $\varphi_k(t)$ and $\psi_k(s)$, where

$$f(t+s) = \sum_{k=0}^{m} \varphi_k(t)\psi_k(s). \tag{2.24}$$

The roles of t and s can be interchanged without loss of generality in (2.24).

Recall that the rule for constructing a trajectory matrix is given by,

$$x_{ij} = f_{i+j-1} \tag{2.25}$$

for $i = 1,\dots,\tau$ and $j = 1,\dots,n(= N-\tau+1)$. Suppose that a time series was sampled, starting at time t, at interval lengths of Δt from a continuous function $f(\cdot)$. If this discretely sampled series is unfolded into a trajectory matrix, then the $(i,j)^{th}$ element of the trajectory matrix is given by,

$$\begin{aligned} x_{ij} &= f_{i+j-1} \\ &= f(t+(i-1)\Delta t+(j-1)\Delta t). \end{aligned} \tag{2.26}$$

for $i = 1,\dots,\tau$ and $j = 1,\dots,n(= N-\tau+1)$.

The above representation makes it possible to write down explicit expressions for the j^{th} column vector of a trajectory matrix.

When a function has an additive property, the Hankel structured trajectory matrix based on a discretely sampled signal, from this function, will be of fixed rank. The additive property of a function also makes it possible to analytically derive base vectors of the r-flat on which the column vectors of the trajectory matrix lie. (see Buchstaber, 1994, p. 7; Golyandina *et al.*, 2001, pp. 239-240)

The following sub-sections will be devoted to exploring continuous functions that possess an additive property

2.4.1 The Class of Power Functions

In this section attention will be given to the continuous function of the form

$$f(t) = a^t, \tag{2.27}$$

where $a > 0$ is an arbitrary constant. This class of continuous functions includes the exponential function as special case, since $a = \exp(1) = 2.7183...$ is simply a constant value.

The class of functions in (2.27) possesses an additive property, since

$$\begin{aligned} f(t+s) &= a^{t+s} \\ &= a^t a^s \\ &= f(t)f(s). \end{aligned} \tag{2.28}$$

Theorem 2.1

If a discrete time series $\{f_t\}_{t=1}^{N}$ is sampled at intervals Δt starting at time t, from the continuous class of power functions

$$f(t) = a^t,\ a > 0, \tag{2.29}$$

then the $\tau \times n$ trajectory matrix $(\mathbf{X})$, formed from such a series, has column vectors belonging to an 1-flat H_1 of rank 1. The following normalised vector spans the 1-flat H_1

$$\mathbf{v}_1 = \mathbf{y} / \|\mathbf{y}\| \tag{2.30}$$

where

$$\mathbf{y} = \begin{bmatrix} a^t & a^{(t+\Delta t)} & \cdots & a^{[t+(\tau-1)\Delta t]} \end{bmatrix}'. \tag{2.31}$$

Proof.

Suppose that a time series $\{f_t\}_{t=1}^N$ of length N was sampled from the continuous function,

$$f(t) = a^t,\ a > 0. \tag{2.32}$$

Suppose that the sample interval was set at Δt, and that the time series was sampled starting at time t. The time series will then be of the form,

$$f_t = \left\{a^t, a^{(t+\Delta t)}, a^{(t+2\Delta t)}, \ldots, a^{[t+(N-1)\Delta t]}\right\}. \tag{2.33}$$

If the series in (2.33) is unfolded into the column vectors of a $\tau \times n$ trajectory matrix, then the following matrix will result

$$\underset{(\tau\times n)}{\mathbf{X}} = \begin{bmatrix} a^{t} & a^{(t+\Delta t)} & a^{(t+2\Delta t)} & \cdots & a^{[t+(n-1)\Delta t]} \\ a^{(t+\Delta t)} & a^{(t+2\Delta t)} & a^{(t+3\Delta t)} & \cdots & a^{[t+n\Delta t]} \\ a^{(t+2\Delta t)} & a^{(t+3\Delta t)} & a^{(t+4\Delta t)} & \cdots & a^{[t+(n+1)\Delta t]} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a^{[t+(\tau-1)\Delta t]} & a^{[t+\tau\Delta t]} & a^{[t+(\tau+1)\Delta t]} & \cdots & a^{[t+(N-1)\Delta t]} \end{bmatrix}.$$

(2.34)

It is clear that the j^{th} column vector in matrix (2.34), can be written as

$$\underset{(\tau\times 1)}{\mathbf{x}_j} = a^{(j-1)\Delta t} \begin{bmatrix} a^{t} \\ a^{(t+\Delta t)} \\ a^{(t+2\Delta t)} \\ \vdots \\ a^{[t+(\tau-1)\Delta t]} \end{bmatrix}. \tag{2.35}$$

But Δt is a constant and so also is a. This implies that $a^{(j-1)\Delta t}$ is also just a constant. Any column vector in the trajectory matrix is therefore a constant multiple of the vector,

$$\underset{(\tau\times 1)}{\mathbf{y}} = \begin{bmatrix} a^{t} \\ a^{(t+\Delta t)} \\ a^{(t+2\Delta t)} \\ \vdots \\ a^{[t+(\tau-1)\Delta t]} \end{bmatrix}. \tag{2.36}$$

This implies that all the column vectors of matrix (2.34) are linearly dependent on the single column vector in (2.36). The trajectory matrix is therefore of rank 1 and so also the 1-flat $H_1 \subset \mathbb{R}^{\tau}, \tau \geq 2$. The normalised vector that spans the 1-flat is therefore $\mathbf{v}_1 = \mathbf{y} / \|\mathbf{y}\|$, with $\mathbf{y}$ as defined in (2.36). This concludes the proof.□

2.4.2 The Class of Sinusoidal Functions

Consider the class of (continuous) sinusoidal functions of the form

$$f(t) = \sin(\varpi t + \varphi) \tag{2.37}$$

where parameter ϖ represents the frequency in radians and φ represents the phase angle.

The class of (continuous) sinusoidal functions also possesses an additive property, viz.

$$\begin{aligned} f(t+s) &= \sin((t+s)\varpi + \varphi) \\ &= f(t)\cos(\varpi s) + f\left(t + \frac{\pi}{2\varpi}\right)\sin(\varpi s). \end{aligned} \tag{2.38}$$

The following theorem illustrates how the trajectory matrix, based on a discretely sampled time series from (2.37), is of rank 2. The theorem also explicitly derives the two spanning vectors of the 2-flat H_2, on which the trajectory matrix column vectors lie.

Theorem 2.2

If a time series $\{f_t\}_{t=1}^{N}$ is sampled at intervals Δt starting at time t, from the continuous class of sinusoidal functions

$$f(t) = \sin(\varpi t + \varphi),$$

then the $\tau \times n$ trajectory matrix $(\mathbf{X})$, formed from such a series, has column vectors belonging to 2-flat H_2 of rank 2. The 2-flat H_2 is spanned by the normalised vectors $\mathbf{v}_1 = \mathbf{y}_1 / \|\mathbf{y}_1\|$ and $\mathbf{v}_2 = \mathbf{y}_2 / \|\mathbf{y}_2\|$, where

$$\begin{aligned} \mathbf{y}_1 &= \left[\cos(\varpi t) \quad \cos(\varpi t + \Delta t\varpi) \quad \cdots \quad \cos[\varpi t + (\tau - 1)\Delta t\varpi]\right]' \\ \mathbf{y}_2 &= \left[\sin(\varpi t) \quad \sin(\varpi t + \Delta t\varpi) \quad \cdots \quad \sin[\varpi t + (\tau - 1)\Delta t\varpi]\right]'. \end{aligned} \tag{2.39}$$

Proof.

Suppose that a discrete time series of length N, was sampled from (2.37) at intervals Δt starting at time t, and is therefore of the form

$$\begin{aligned} f_t &= \{f(t), f(t+\Delta t), \ldots, f(t+(N-1)\Delta t)\} \\ &= \{\sin(\varpi t+\varphi), \sin[\varpi(t+\Delta t)+\varphi], \ldots, \sin[\varpi(t+[N-1]\Delta t)+\varphi]\}. \end{aligned} \tag{2.40}$$

The following equation relates the $(i,j)^{th}$ element in the trajectory matrix directly to the continuous function from which it was sampled,

$$\begin{aligned} x_{ij} &= f_{i+j-1} \\ &= f(t+(i+j-2)\Delta t) \\ &= f(t+(i-1)\Delta t+(j-1)\Delta t) \\ &= \sin([t+(i-1)\Delta t+(j-1)\Delta t]\varpi+\varphi) \\ &= \sin(\underbrace{\varpi t+(i-1)\varpi\Delta t}_{B}+\underbrace{(j-1)\varpi\Delta t+\varphi}_{A}) \end{aligned} \tag{2.41}$$

for $i = 1,\ldots,\tau$ and $j = 1,\ldots,n(= N-\tau+1)$. The following standard trigonometric result can then be invoked,

$$\sin(A+B) = \sin(A)\cos(B)+\cos(A)\sin(B). \tag{2.42}$$

Combining (2.41) and (2.42), results in

$$\begin{aligned} x_{ij} &= \sin((j-1)\varpi\Delta t+\varphi)\cos(\varpi t+(i-1)\varpi\Delta t) \\ &\quad +\cos((j-1)\varpi\Delta t+\varphi)\sin(\varpi t+(i-1)\varpi\Delta t). \end{aligned} \tag{2.43}$$

From (2.43) it is easy to find the j^{th} column vector of the trajectory matrix

$$\mathbf{x}_j = \begin{bmatrix} \{\sin((j-1)\varpi\Delta t+\varphi)\cos(\varpi t) \\ +\cos((j-1)\varpi\Delta t+\varphi)\sin(\varpi t)\} \\ \{\sin((j-1)\varpi\Delta t+\varphi)\cos(\varpi t+\varpi\Delta t) \\ +\cos((j-1)\varpi\Delta t+\varphi)\sin(\varpi t+\varpi\Delta t)\} \\ \vdots \\ \{\sin((j-1)\varpi\Delta t+\varphi)\cos(\varpi t+(\tau-1)\varpi\Delta t) \\ +\cos((j-1)\varpi\Delta t+\varphi)\sin(\varpi t+(\tau-1)\varpi\Delta t)\} \end{bmatrix}. \tag{2.44}$$

In (2.44) the following are constants: ϖ since it is the frequency of the sinusoid, φ since it is the phase angle of the sinusoid and the column vector index j. It is therefore clear that $\sin(\varpi(j-1)\varpi\Delta t+\varphi)$ and $\cos(\varpi(j-1)\varpi\Delta t+\varphi)$ are both constant. The j^{th} column vector in the trajectory matrix will therefore be a linear combination of the vectors

$$\begin{aligned} \mathbf{y}_1 &= \left[\cos(\varpi t) \quad \cos(\varpi t+\Delta t\varpi) \quad \cdots \quad \cos[\varpi t+(\tau-1)\Delta t\varpi]\right]' \\ \mathbf{y}_2 &= \left[\sin(\varpi t) \quad \sin(\varpi t+\Delta t\varpi) \quad \cdots \quad \sin[\omega t+(\tau-1)\Delta t\varpi]\right]'. \end{aligned} \tag{2.45}$$

Since any column vector is a linear combination of these two linearly independent vectors, the trajectory matrix is of rank 2 and so also the 2-flat $H_2 \subset \mathbb{R}^\tau, \tau \geq 3$ on which the column vectors fall. The normalised vectors that span the 2-flat is given by $\mathbf{v}_1 = \mathbf{y}_1/\|\mathbf{y}_1\|$ and $\mathbf{v}_2 = \mathbf{y}_2/\|\mathbf{y}_2\|$, with $\mathbf{y}_1$ and $\mathbf{y}_2$ as defined in (2.45). This concludes the proof. □

2.4.3 The Class of Polynomial Functions

Consider the class of continuous polynomial functions of order m

$$\begin{aligned} f(t) &= \sum_{k=0}^{m} b_k t^k \\ &= b_0 + b_1 t^1 + b_2 t^2 + \ldots + b_m t^m. \end{aligned} \tag{2.46}$$

The following theorem is used to show how a discretely sampled series from (2.46) has a trajectory matrix of rank $(m+1)$.

Theorem 2.3

If a time series $\{f_t\}_{t=1}^{N}$ is sampled at intervals Δt starting at time t, from the continuous class of polynomials of order m

$$f(t)=\sum_{k=0}^{m} b_k t^k, \tag{2.47}$$

then the $\tau \times n$ trajectory matrix $(\mathbf{X})$, formed from such a series, has column vectors belonging to the $(m+1)$-flat H_{m+1} of rank $(m+1)$. The $(m+1)$-flat H_{m+1} is spanned by the normalised vectors $\mathbf{v}_1=\mathbf{y}_1/\|\mathbf{y}_1\|, \mathbf{v}_2=\mathbf{y}_2/\|\mathbf{y}_2\|,\ldots,\mathbf{v}_{m+1}=\mathbf{y}_{m+1}/\|\mathbf{y}_{m+1}\|$, where

$$\begin{aligned}\mathbf{y}_1 &= \begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}' \\ \mathbf{y}_k &= \begin{bmatrix}\dfrac{t^{k-1}}{(k-1)!} & \dfrac{(t+\Delta t)^{k-1}}{(k-1)!} & \cdots & \dfrac{[t+(\tau-1)\Delta t]^{k-1}}{(k-1)!}\end{bmatrix}' \quad for\ k=2,\ldots,m+1.\end{aligned}$$

(2.48)

Proof.

Let the k^{th}-order derivative of a continuous function $f(t)$, with respect to t, be denoted by

$$f^{(k)}(t)\overset{def}{=}\frac{d^k}{dt^k}f(t). \tag{2.49}$$

Using a Taylor series expansion, it follows that

$$f(t+s)=\sum_{k=0}^{m} f^{(k)}(t)\frac{s^k}{k!}. \tag{2.50}$$

Suppose now that a series of length N was sampled from (2.46) at interval length Δt starting at time t. Suppose, therefore, that the series is given by $\{f_t\}_{t=1}^{N}=\{f(t), f(t+\Delta t),\ldots, f(t+(N-1)\Delta t)\}$.

Using the result in (2.50), it follows that the following equation relates the $(i,j)^{th}$ element in the trajectory matrix directly to the continuous function from which it was sampled

$$\begin{aligned} x_{ij} &= f_{i+j-1} \\ &= f\left(\underbrace{t+(i-1)\Delta t}_{s}+\underbrace{(j-1)\Delta t}_{t}\right) \\ &= \sum_{k=0}^{m} f^{(k)}((j-1)\Delta t)\frac{[t+(i-1)\Delta t]^k}{k!}. \end{aligned} \tag{2.51}$$

The roles of t and s were interchanged in (2.50), without loss of generality, to arrive at (2.51).

Using (2.51) it now follows that the j^{th} column vector of the trajectory matrix is given by

$$\mathbf{x}_j=\begin{bmatrix} \sum_{k=0}^{m} f^{(k)}((j-1)\Delta t)\frac{(t)^k}{k!} \\ \sum_{k=0}^{m} f^{(k)}((j-1)\Delta t)\frac{(t+\Delta t)^k}{k!} \\ \vdots \\ \sum_{k=0}^{m} f^{(k)}((j-1)\Delta t)\frac{[t+(\tau-1)\Delta t]^k}{k!} \end{bmatrix}. \tag{2.52}$$

On closer inspection of (2.52), it follows that the constant term $f^{(m)}((j-1)\Delta t)$ is included in the sum $\sum_{k=0}^{m} f^{(k)}((j-1)\Delta t)$, where

$$f^{(m)}((j-1)\Delta t) = \frac{d^m}{dt^m}\sum_{k=0}^{m} b_k t^k = m!b_m . \tag{2.53}$$

Hence, the j^{th} column vector of the trajectory matrix can be written as

$$\mathbf{x}_j = \begin{bmatrix} \sum_{k=0}^{m-1} f^{(k)}((j-1)\Delta t)\frac{(t)^k}{k!} + f^{(m)}((j-1)\Delta t)\frac{(t)^m}{m!} \\ \sum_{k=0}^{m-1} f^{(k)}((j-1)\Delta t)\frac{(t+\Delta t)^k}{k!} + f^{(m)}((j-1)\Delta t)\frac{(t+\Delta t)^m}{m!} \\ \vdots \\ \sum_{k=0}^{m-1} f^{(k)}((j-1)\Delta t)\frac{[t+(\tau-1)\Delta t]^k}{k!} + f^{(m)}((j-1)\Delta t)\frac{[t+(\tau-1)\Delta t]^m}{m!} \end{bmatrix}$$
$$= \begin{bmatrix} \sum_{k=0}^{m-1} f^{(k)}((j-1)\Delta t)\frac{(t)^k}{k!} + m!b_m\frac{(t)^m}{m!} \\ \sum_{k=0}^{m-1} f^{(k)}((j-1)\Delta t)\frac{(t+\Delta t)^k}{k!} + m!b_m\frac{(t+\Delta t)^m}{m!} \\ \vdots \\ \sum_{k=0}^{m-1} f^{(k)}((j-1)\Delta t)\frac{[t+(\tau-1)\Delta t]^k}{k!} + m!b_m\frac{[t+(\tau-1)\Delta t]^m}{m!} \end{bmatrix}. \tag{2.54}$$

Let now

$$\mathbf{y}_1 = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}'$$
$$\mathbf{y}_k = \begin{bmatrix} \frac{t^{k-1}}{(k-1)!} & \frac{(t+\Delta t)^{k-1}}{(k-1)!} & \cdots & \frac{[t+(\tau-1)\Delta t]^{k-1}}{(k-1)!} \end{bmatrix}' \quad for\ k = 2,\ldots,m+1. \tag{2.55}$$

It is clear that the vectors in (2.55) are linearly independent, and that the j^{th} column vector of the trajectory matrix can be written as a linear combination of the vectors featuring in (2.55), viz.

$$\mathbf{x}_j = \sum_{k=0}^{m-1} f^{(k)}((j-1)\Delta t)\mathbf{y}_{k+1} + m! b_m \mathbf{y}_{m+1}. \tag{2.56}$$

The trajectory matrix is therefore of rank $(m+1)$ and so also the $(m+1)$-flat $H_{m+1} \subset \mathbb{R}^{\tau}, \tau \geq m+2$. The $(m+1)$-flat H_{m+1} is spanned by the normalised base vectors $\mathbf{v}_1 = \mathbf{y}_1 / \|\mathbf{y}_1\|, \mathbf{v}_2 = \mathbf{y}_2 / \|\mathbf{y}_2\|, \ldots, \mathbf{v}_{m+1} = \mathbf{y}_{m+1} / \|\mathbf{y}_{m+1}\|$, with $\mathbf{y}_1, \ldots, \mathbf{y}_{m+1}$ as defined in (2.55). This concludes the proof. □

2.4.4 A General Class of Functions

Buchstaber (1994, pp. 7-8) proposed the following general class of functions that possesses an additive property

$$f(t) = \sum_{k=1}^{K} p_k(t) \exp(\lambda_k t) \sin(\varpi_k t + \varphi_k) \tag{2.57}$$

where $p_k(t)$ are polynomials.

It is clear from (2.57) that the unfolding into a trajectory matrix can handle a rather broad class of functions. The class of functions in (2.57) is able to handle a wide spectrum of patterns that are generally found in time series, viz.

- Linear trend
- Quadratic trend
- Monthly seasonal period without trend (sinusoid with period 12)
- Combination of Linear trend and additive seasonal pattern (first-order polynomial combined with a sinusoid)
- Combination of Linear trend and multiplicative seasonal pattern (first-order polynomial combined with a sinusoid)
- etc.

Series that also fall into the class defined in (2.57) involve the autocorrelation function (ACF), formed using observed time series from the broad ARIMA class processes. It is known that the class of autoregressive processes (AR(p)) has an ACF that exhibits patterns that die down to zero as the lag-time increases, either as patterns including exponential decay and/or damped sinusoidal patterns.

2.5 Linear Recurrent Formulae

A more recent (Golyandina *et al.*, 2001) development in SSA uses the notion of linear recurrent formulae (LRF). The definition of an LRF follows.

Definition 2.6 Linear Recurrent Formula (LRF)

A *linear recurrent formula (LRF)* is defined by

$$f_{t+r} = \sum_{i=1}^{r} a_i f_{t+r-i} \,, \tag{2.58}$$

for $t = 1,\ldots,N-r+1$ and $a_r \neq 0$. In (2.58), r is known as the *finite-difference dimension* of the series $\{f_t\}_{t=1}^{N}$. The broad class of continuous functions in (2.57), proposed by Buchstaber (1994), satisfies the property in (2.58). It is therefore more general to view SSA in the light of LRF, as it is clear from Golyandina *et al.* (2001) that SSA can handle functions governed by LRF. The benefit of viewing SSA through the spectacle of linear recurrent formulae is that coefficients $(a_1,\ldots,a_r)$ can be estimated in the process. These coefficients usually possess certain characteristics, e.g. lying inside a unit circle. Golyandina *et al.* (2001) can be consulted for this approach.

It is interesting to note that the class of power functions in §2.4.1 is, for example, governed by the LRF

$$f_t = a f_{t-1}. \tag{2.59}$$

According to Golyandina *et al.* (2001, p. 97) the function $f(t) = \cos(2\pi\omega t + \varphi)$ is governed by the LRF

$$f_t = 2\cos(2\pi\varpi) f_{t-1} - f_{t-2}. \tag{2.60}$$

It is also important to note that a series $\{f_t\}_{t=1}^{N}$, governed by an LRF of dimension $r \geq 1$, has $(j+r)^{th}$ trajectory matrix column vector,

$$\mathbf{x}_{j+r} = \sum_{i=1}^{r} a_i \mathbf{x}_{j+r-i},\ j = 1,\ldots,(n-r) \tag{2.61}$$

which is clearly a linear combination of r column vectors of the trajectory matrix, implying that the column dimension is at most r.

2.6 Noise Models and Statistical Attributes

In this section the addition of noise to a discretely sampled time series, from a function with an additive property, defined in (2.24), will be considered. Two noise structures, viz. white noise and first-order serially correlated noise, will be entertained. This study of noise structures attempts to investigate statistical attributes of SSA. The section also lays the foundation for Chapter 4 and Chapter 5 that deal with model selection and inferential issues.

Observed time series, in general, contain noise. A "signal series" is usually hypothesised as belonging to a certain class of functions or of probabilistic models, and inherent noise hypothesised to originate from some statistical distribution. According to Shumway and Stoffer (2000) p. 15,

"*The problems of detecting a signal and then in estimating or extracting the waveform* x_t *are of great interest in many areas of engineering and the physical and biological sciences. In economics, the underlying signal may be trend or it may be a seasonal component of a series.*"

According to them noise can be "*white or correlated over time*". The first of these two possibilities is thought to occur more frequently in engineering applications. The latter of these two situations is expected to occur frequently in economic time series. Economic time series tend to have a significant first-order correlation structure. This is intuitively sound, as a time series value at time t should have a large amount of "information" inherited from the observation at time $t-1$.

The case of additive white noise will now be considered in detail.

2.6.1 The Additive White Noise Assumption

In this section it is assumed that an observed time series $\{x_t\}_{t=1}^{N}$ originates from the following model,

$$X_t = \underset{\text{signal}}{f_t} + \underset{\text{noise}}{\varepsilon_t} \ , \ t = 1,\dots,N \tag{2.62}$$

where $\{\varepsilon_t\}_{t=1}^{N}$ is a white noise series with distributional assumption

$$\varepsilon_t \overset{iid}{\sim} N\left(0,\ \sigma^2\right), \tag{2.63}$$

and $\{f_t\}_{t=1}^{N}$ is a signal series from the broad class of functions in (2.57), as introduced by Buchstaber (1994).

If $\{f_t\}_{t=1}^{N}$ was sampled from a polynomial of any order, then the model given by (2.62) is actually a polynomial regression model (Bowerman and O'Connell, 1993). The form of model in (2.62) is, in general, a time series regression model (Makridakis *et al.*, 1998, p.187). This, unfortunately, does not imply that the polynomial coefficients can be estimated using maximum likelihood or ordinary least squares. SSA is a non-parametric method, and it is assumed that no knowledge of the exact structural form of the functions from which the signal series was sampled, is available. Under such an assumption, it is impossible to utilise the usual parameter estimation methods and a non-parametric method is required.

It is clear from the assumptions that,

$$X_t \sim N\left(f_t,\ \sigma^2\right). \tag{2.64}$$

The time series literature states that there can be immense problems with, *inter alia,* parameter estimation, when a time series is a realization from the following type of stochastic process,

$$X_t \sim N\left(\mu_t,\ \sigma^2\right). \tag{2.65}$$

This situation makes estimation of the mean level (μ_t) of a time series difficult, since only a single realization $(x_t,\ t = 1,\ldots,N)$ per time dependent variable $(X_t,\ t = 1,\ldots,N\ )$ is usually available. To overcome this problem, the concept of a stationary time series is usually introduced (Harvey, 1993; Granger and Newbold, 1986). A weak second-order stationary stochastic process is defined to have a time independent mean level (μ) and time independent variance σ^2. This "assumption" makes sensible estimation possible. Finding the joint distribution of a stochastic process can still be a daunting task, even in the case of weak second-order stationarity.

In the current context, the above reasoning cannot be resorted to. A single time series realization is available, in the form of $(x_t,\ t = 1,\ldots,N)$. As mentioned, each observation (x_t) originates from a variable (X_t) that is distributed as $N\left(\mu_t,\ \sigma^2\right)$. It is clear that a single realization for the mean level μ_t is available in the form of f_t. This effectively makes estimation of the mean level impossible, in the context of general statistical theory.

Unfolding an observed time series from (2.62) into the column vectors of a trajectory matrix, places a univariate time series in a multivariate framework. The multivariate statistical distributional form is of great importance here. It immediately becomes clear from such knowledge why robust multivariate techniques, based on M-estimators, do not apply in the context of SSA. This will be further elaborated upon in Chapter 4, as it influences the choice of robust multivariate statistical techniques, used during model selection. This study also shows why many robust multivariate statistical techniques do not find application in the context of a trajectory matrix.

To study the multivariate distributional properties of an unfolding of an observed time series from the class model defined in (2.62), consider the following unfolding

$$\underset{(\tau\times n)}{\mathbf{X}} = \underset{(\tau\times n)}{\mathbf{F}} + \underset{(\tau\times n)}{\mathbf{E}}$$

$$\begin{bmatrix} X_1 & X_2 & \cdots & X_n \\ X_2 & X_3 & \cdots & X_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ X_\tau & X_{\tau+1} & \cdots & X_N \end{bmatrix} = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \\ f_2 & f_3 & \cdots & f_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ f_\tau & f_{\tau+1} & \cdots & f_N \end{bmatrix} + \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \cdots & \varepsilon_n \\ \varepsilon_2 & \varepsilon_3 & \cdots & \varepsilon_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_\tau & \varepsilon_{\tau+1} & \cdots & \varepsilon_N \end{bmatrix} \tag{2.66}$$

$$\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \cdots & \mathbf{f}_n \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon}_1 & \boldsymbol{\varepsilon}_2 & \cdots & \boldsymbol{\varepsilon}_n \end{bmatrix}$$

where

$\mathbf{X}$ = Hankel structured trajectory matrix of observed time series values
$\mathbf{F}$ = Hankel structured trajectory matrix of signal series observations
$\mathbf{E}$ = Hankel structured trajectory matrix of noise observations.

The column vectors of trajectory matrix $\mathbf{F}$ fall on a single r-flat H_r. This is due to the assumption that the column vectors represent an unfolding of the series $\{f_t\}_{t=1}^{N}$, assumed to originate from (2.57). The column vectors of trajectory matrix $\mathbf{X}$ do not fall on a single r-flat, but represent a multivariate scatter around the r-flat H_r. This trajectory matrix is based on the white noise-distorted series $\{x_t\}_{t=1}^{N}$. Trajectory matrix $\mathbf{E}$ contains the stochastic component of the white-noise distorted series. The column vectors of this matrix are therefore stochastic. The distribution of a typical column vector $\left(\boldsymbol{\varepsilon}_j\right)$ must therefore be derived.

Consider the j^{th} column vector in trajectory matrix $\mathbf{E}$, where

$$\underset{(\tau\times 1)}{\boldsymbol{\varepsilon}_j} = \begin{bmatrix} \varepsilon_j \\ \varepsilon_{j+1} \\ \vdots \\ \varepsilon_{j+\tau-1} \end{bmatrix}. \tag{2.67}$$

It is clear from the assumptions that the column vectors of matrix $\mathbf{E}$ are identically distributed according to a multivariate normal distribution, viz.

$$\underset{(\tau\times 1)}{\boldsymbol{\varepsilon}_j} \sim N_\tau\left(\mathbf{0},\, \sigma^2\mathbf{I}_\tau\right). \tag{2.68}$$

The column vectors in the stochastic trajectory matrix $\mathbf{E}$ are, however, not independent. The covariance between arbitrary column vectors $\boldsymbol{\varepsilon}_k$ and $\boldsymbol{\varepsilon}_l$ is given by,

$$\begin{aligned} \underset{(\tau\times\tau)}{\Sigma_{\boldsymbol{\varepsilon}_k\boldsymbol{\varepsilon}_l}} &= Cov\left(\boldsymbol{\varepsilon}_k,\, \boldsymbol{\varepsilon}_l'\right) \\ &= \begin{bmatrix} \underset{(|k-l|\times\tau-|k-l|)}{\mathbf{0}} & \underset{(|k-l|\times|k-l|)}{\mathbf{0}} \\ \sigma^2\mathbf{I}_{\tau-|k-l|} & \underset{(\tau-|k-l|\times|k-l|)}{\mathbf{0}} \end{bmatrix} \end{aligned} \tag{2.69}$$

where

$$k = 1,\ldots,n(= N-\tau+1)\ ;\ l = 1,\ldots,n(= N-\tau+1)\ ;\ |k-l| \leq \tau - 1.$$

It follows directly that the j^{th} column vector of stochastic trajectory matrix $\mathbf{X}$, is multivariate normally distributed, with

$$\underset{(\tau\times1)}{\mathbf{X}_j} \sim N_\tau\left(\mathbf{f}_j,\ \sigma^2\mathbf{I}_\tau\right). \tag{2.70}$$

The column vectors are therefore not identically distributed and are also not independent. This is an extremely important fact. The covariance between arbitrary column vectors $\mathbf{X}_k$ and $\mathbf{X}_l$ is of course the same as between $\mathbf{e}_k$ and $\mathbf{e}_l$ and thus from (2.69) given by

$$\begin{aligned}\underset{(\tau\times\tau)}{\Sigma_{\mathbf{X}_k\mathbf{X}_l}} &= Cov\left(\mathbf{X}_k,\ \mathbf{X}_l'\right)\\ &= \begin{bmatrix} \underset{(|k-l|\times\tau-|k-l|)}{\mathbf{0}} & \underset{(|k-l|\times|k-l|)}{\mathbf{0}} \\ \sigma^2\mathbf{I}_{\tau-|k-l|} & \underset{(\tau-|k-l|\times|k-l|)}{\mathbf{0}} \end{bmatrix}\end{aligned} \tag{2.71}$$

where

$$k = 1,\ldots,n(= N-\tau+1)\ ;\ l = 1,\ldots,n(= N-\tau+1)\ ;\ |k-l| \le \tau-1.$$

The model with a serially correlated noise structure is considered in §2.6.2 below

2.6.2 The First-Order Serially Correlated Noise Assumption

In this section it is assumed that an observed time series $\{x_t\}_{t=1}^{N}$ originates from the following model

$$X_t = f_t + \varepsilon_t\ ,\ t = 1,\ldots,N\ , \tag{2.72}$$

where the noise process is first-order serially correlated such that,

$$\varepsilon_t = \phi_1\varepsilon_{t-1} + Z_t\ ,\ t = 1,\ldots,N \tag{2.73}$$

and

$$Z_t \overset{iid}{\sim} N\left(0,\ \sigma^2\right). \tag{2.74}$$

It is clear from (2.73) that the noise follows an AR(1)-process. It therefore has expectation

$$E\left(\varepsilon_t\right) = 0, \tag{2.75}$$

and covariance

$$Cov\left(\varepsilon_t,\ \varepsilon_{t+k}\right) = \frac{\sigma^2}{1-\phi_1^2}\phi_1^{|k|}. \tag{2.76}$$

From (2.72) and (2.73) it follows that

$$X_t \sim N\left(f_t,\ \frac{\sigma^2}{1-\phi_1^2}\right). \tag{2.77}$$

The random variables $\left\{X_t,\ t = 1,\ldots,N\right\}$ are therefore not identically distributed and are not independent. The covariance between X_t and X_{t+k} is given by,

$$Cov\left(X_t,\ X_{t+k}\right) = \frac{\sigma^2}{1-\phi_1^2}\phi_1^{|k|}. \tag{2.78}$$

The unfolding of the model in (2.72) into a trajectory matrix is considered next, viz.

$$\begin{gathered}
\mathbf{X} = \mathbf{F} + \mathbf{E} \\
\begin{bmatrix} X_1 & X_2 & \cdots & X_n \\ X_2 & X_3 & \cdots & X_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ X_\tau & X_{\tau+1} & \cdots & X_N \end{bmatrix} = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \\ f_2 & f_3 & \cdots & f_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ f_\tau & f_{\tau+1} & \cdots & f_N \end{bmatrix} + \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \cdots & \varepsilon_n \\ \varepsilon_2 & \varepsilon_3 & \cdots & \varepsilon_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_\tau & \varepsilon_{\tau+1} & \cdots & \varepsilon_N \end{bmatrix} \\
\left[\mathbf{X}_1\ \ \mathbf{X}_2\ \ \cdots\ \ \mathbf{X}_n\right] = \left[\mathbf{f}_1\ \ \mathbf{f}_2\ \ \cdots\ \ \mathbf{f}_n\right] + \left[\boldsymbol{\varepsilon}_1\ \ \boldsymbol{\varepsilon}_2\ \ \cdots\ \ \boldsymbol{\varepsilon}_n\right]
\end{gathered} \tag{2.79}$$

where

$$\mathbf{X} = \text{Hankel structured trajectory matrix of observed time series values}$$
$$\mathbf{F} = \text{Hankel structured trajectory matrix of signal series observations}$$
$$\mathbf{E} = \text{Hankel structured trajectory matrix of noise observations.}$$

A typical column vector of the stochastic matrix $\mathbf{E}$ will have the form

$$\underset{(\tau\times 1)}{\boldsymbol{\varepsilon}_j} = \begin{bmatrix} \varepsilon_j \\ \varepsilon_{j+1} \\ \vdots \\ \varepsilon_{j+\tau-1} \end{bmatrix}. \tag{2.80}$$

The expected value of the above column vector is given by

$$E\left(\boldsymbol{\varepsilon}_j\right) = \begin{bmatrix} E\left(\varepsilon_j\right) \\ E\left(\varepsilon_{j+1}\right) \\ \vdots \\ E\left(\varepsilon_{j+\tau-1}\right) \end{bmatrix} = \mathbf{0}_\tau, \tag{2.81}$$

and the covariance matrix is given by

$$\underset{(\tau\times\tau)}{\boldsymbol{\Sigma}} = Cov\left(\boldsymbol{\varepsilon}_j,\ \boldsymbol{\varepsilon}_j'\right)$$
$$= \begin{bmatrix} E\left(\varepsilon_j^2\right) & E\left(\varepsilon_j\varepsilon_{j+1}\right) & \cdots & E\left(\varepsilon_j\varepsilon_{j+\tau-1}\right) \\ E\left(\varepsilon_{j+1}\varepsilon_j\right) & E\left(\varepsilon_{j+1}^2\right) & \cdots & E\left(\varepsilon_{j+1}\varepsilon_{j+\tau-1}\right) \\ \vdots & \vdots & \ddots & \cdots \\ E\left(\varepsilon_{j+\tau-1}\varepsilon_j\right) & E\left(\varepsilon_{j+\tau-1}\varepsilon_{j+1}\right) & \cdots & E\left(\varepsilon_{j+\tau-1}^2\right) \end{bmatrix}$$
$$= \frac{\sigma^2}{1-\phi_1^2}\begin{bmatrix} 1 & \phi_1^1 & \cdots & \phi_1^{\tau-1} \\ \phi_1^1 & 1 & \cdots & \phi_1^{\tau-2} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1^{\tau-1} & \phi_1^{\tau-2} & \cdots & 1 \end{bmatrix}. \tag{2.82}$$

Hence, the column vectors are multivariate normally distributed as follows

$$\boldsymbol{\varepsilon}_j \sim N_\tau\left(\mathbf{0},\ \boldsymbol{\Sigma}\right). \tag{2.83}$$

The column vectors in stochastic matrix $\mathbf{E}$ are, however, not independent. The covariance between arbitrary column vectors $\boldsymbol{\varepsilon}_k$ and $\boldsymbol{\varepsilon}_l$ is given by

$$\underset{(\tau\times\tau)}{\Sigma_{\boldsymbol{\varepsilon}_k\boldsymbol{\varepsilon}_l}} = Cov\left(\boldsymbol{\varepsilon}_k,\ \boldsymbol{\varepsilon}_l'\right)$$
$$= \begin{bmatrix} E\left(\varepsilon_k\varepsilon_l\right) & E\left(\varepsilon_k\varepsilon_{l+1}\right) & \cdots & E\left(\varepsilon_k\varepsilon_{l+\tau-1}\right) \\ E\left(\varepsilon_{k+1}\varepsilon_l\right) & E\left(\varepsilon_{k+1}\varepsilon_{l+1}\right) & \cdots & E\left(\varepsilon_{k+1}\varepsilon_{l+\tau-1}\right) \\ \vdots & \vdots & \ddots & \cdots \\ E\left(\varepsilon_{k+\tau-1}\varepsilon_l\right) & E\left(\varepsilon_{k+\tau-1}\varepsilon_{l+1}\right) & \cdots & E\left(\varepsilon_{k+\tau-1}\varepsilon_{l+\tau-1}\right) \end{bmatrix}$$
$$= \frac{\sigma^2}{1-\phi_1^2}\begin{bmatrix} \phi_1^{|k-l|} & \phi_1^{|k-l-1|} & \cdots & \phi_1^{|k-l-\tau+1|} \\ \phi_1^{|k-l-1|} & \phi_1^{|k-l|} & \cdots & \phi_1^{|k-l-\tau+2|} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1^{|k-l-\tau+1|} & \phi_1^{|k-l-\tau+2|} & \cdots & \phi_1^{|k-l|} \end{bmatrix}. \tag{2.84}$$

Consider a typical column vector of matrix $\mathbf{X}$

$$\underset{(\tau\times 1)}{\mathbf{X}_j} = \begin{bmatrix} X_j \\ X_{j+1} \\ \vdots \\ X_{j+\tau-1} \end{bmatrix} = \begin{bmatrix} f_j \\ f_{j+1} \\ \vdots \\ f_{j+\tau-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_j \\ \varepsilon_{j+1} \\ \vdots \\ \varepsilon_{j+\tau-1} \end{bmatrix}. \tag{2.85}$$

The expected value of the column vector is given by

$$E\left(\mathbf{X}_j\right) = \mathbf{f}_j \tag{2.86}$$

and the covariance matrix is given by

$$\begin{aligned} \underset{(\tau\times\tau)}{\boldsymbol{\Sigma}} &= Cov\left(\mathbf{X}_j,\ \mathbf{X}_j'\right) \\ &= \frac{\sigma^2}{1-\phi_1^2}\begin{bmatrix} 1 & \phi_1^1 & \cdots & \phi_1^{\tau-1} \\ \phi_1^1 & 1 & \cdots & \phi_1^{\tau-2} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1^{\tau-1} & \phi_1^{\tau-2} & \cdots & 1 \end{bmatrix}. \end{aligned} \tag{2.87}$$

The column vector $\mathbf{X}_j$ is therefore multivariate normally distributed, with

$$\mathbf{X}_j \sim N_\tau\left(\mathbf{f}_j,\ \boldsymbol{\Sigma}\right). \tag{2.88}$$

The column vectors of the trajectory matrix are therefore not identically distributed and are also not independent. The covariance between arbitrary column vectors $\mathbf{X}_k$ and $\mathbf{X}_l$ is given by

$$\underset{(\tau\times\tau)}{\boldsymbol{\Sigma}_{\mathbf{X}_k\mathbf{X}_l}} = Cov\left(\mathbf{X}_k\,,\ \mathbf{X}_l'\right)$$
$$= \frac{\sigma^2}{1-\phi_1^2}\begin{bmatrix} \phi_1^{|k-l|} & \phi_1^{|k-l-1|} & \cdots & \phi_1^{|k-l-\tau+1|} \\ \phi_1^{|k-l-1|} & \phi_1^{|k-l|} & \cdots & \phi_1^{|k-l-\tau+2|} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1^{|k-l-\tau+1|} & \phi_1^{|k-l-\tau+2|} & \cdots & \phi_1^{|k-l|} \end{bmatrix}, \qquad (2.89)$$

where

$$k = 1,...,n(= N-\tau+1)\ ;\ l = 1,...,n(= N-\tau+1)\ ;\ |k-l| \leq \tau - 1.$$

The above results are of crucial importance from a statistical point of view, and must be kept in mind. Singular value decomposition will be considered in §2.7 below.

2.7 Singular Value Decomposition (SVD) in SSA

2.7.1 Basics and Definitions

Singular Value Decomposition (SVD) (Gentle, 1998; Gill *et al.*, 1991; Trefethen and Bau, 1997) and Principal Component Analysis (PCA) (Joliffe, 1986; Scott, 1992) are well-known dimension reduction techniques, and are considered in standard literature on numeric algebra and multivariate statistical analysis. Principal component analysis is used to represent a set of variables with a new, smaller, set of variables that are mutually orthogonal. PCA uses SVD to obtain eigenvalues and corresponding eigenvectors. The actual principal components are constructed using the eigenvectors obtained from SVD. This section will explore the use of SVD as dimension reduction technique in the context of SSA.

A very thorough treatment on SVD can be found in Chapter 4 of Golyandina *et al.* (2001). The particular Chapter is essential for understanding the usefulness of SVD in the SSA context.

It is also important to point out statistical issues when applying SVD as dimension reduction technique. According to Jackson (1991, p. 365)

"*...the estimates of the characteristic vectors, themselves, are distribution-free although inferential procedures associated with them are not.*"

It is therefore clear that PCA is a distribution-free technique. The same argument holds for SVD as dimension reduction technique. This is an important consideration, as it was pointed out in §2.6, cf. (2.70) and (2.88), that the multivariate distributional theory underlying SSA is rather complex.

It was shown that the column vectors of the trajectory matrix were not independent and identically distributed. Under the normality assumption, each of the column vectors has its own multivariate normal distribution, each with its own vector of means $\left(\mathbf{f}_j\right)$, cf. (2.70) and (2.88) in §2.6 above. To worsen matters, there exists only a single realization per multivariate normal distribution in the form of a single column vector. Also, each column vector in the trajectory matrix has $(\tau - 1)$ elements in common, with the column vector immediately to its left. This causes a rather peculiar correlation structure.

From the above argument it is clear that care should be used when applying statistical techniques in the SSA context. Fortunately, SVD is a distribution-free technique and the above arguments do not influence the use thereof.

A formal definition of SVD is now given.

Definition 2.7 Singular Value Decomposition (SVD) of a Matrix

The *singular value decomposition (SVD)* of a $\tau \times n$ matrix $\mathbf{X}$ of rank r, is defined by

$$\mathbf{X} = \mathbf{V}\boldsymbol{\Lambda}^{1/2}\mathbf{U}' \tag{2.90}$$

or alternatively by

$$\underset{(\tau\times n)}{\mathbf{X}} = \sum_{j=1}^{r} \sqrt{\lambda_j}\mathbf{v}_j\mathbf{u}'_j , \tag{2.91}$$

where

$$\begin{aligned}
\underset{(\tau\times r)}{\mathbf{V}} &= \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_r \end{bmatrix} \\
\underset{(n\times r)}{\mathbf{U}} &= \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \end{bmatrix} \\
\underset{(\tau\times\tau)}{\mathbf{\Lambda}^{1/2}} &= diag\left(\sqrt{\lambda_1},\sqrt{\lambda_2},\ldots,\sqrt{\lambda_r}\right) \\
\mathbf{V}'\mathbf{V} &= \mathbf{I}_r \\
\mathbf{U}'\mathbf{U} &= \mathbf{I}_r .
\end{aligned} \tag{2.92}$$

$\mathbf{V}$ is the matrix of left singular (eigen-) vectors of matrix $\mathbf{X}$, and $\mathbf{U}$ is the matrix of right singular (eigen-) vectors of matrix $\mathbf{X}$ and $r \leq \min(\tau, n)$. Furthermore, $\mathbf{\Lambda}^{1/2} = diag\left(\sqrt{\lambda_1},\sqrt{\lambda_2},\ldots,\sqrt{\lambda_r}\right)$ is a diagonal matrix with diagonal elements the singular (eigen-) values of matrix $\mathbf{X}$. It is convention to consider the eigenvalues in order of magnitude, i.e. letting $\sqrt{\lambda_1} \geq \sqrt{\lambda_2} \geq \cdots \geq \sqrt{\lambda_r} > 0$ and $\sqrt{\lambda_{r+1}} = \sqrt{\lambda_{r+2}} = \ldots = \sqrt{\lambda_\tau} = 0$.

The form in (2.91) is also known as the *spectral decomposition of a matrix*. This is also where singular spectrum analysis (SSA) inherits its name. The first reference to baptize as SSA the methodology of unfolding a matrix into a trajectory matrix and performing SVD, dates back to the work by Vautard and Ghil (1989). The eigenvectors derived through the use of SVD are sometimes also referred to as empirical orthogonal functions (EOFs).

Theorem 2.4 below relates the eigenvalues and eigenvectors of the $\tau \times n$ trajectory matrix $\mathbf{X}$ to the $\tau \times \tau$ matrix $\mathbf{XX}'$. The column vectors of the trajectory matrix $\mathbf{X}$ form an over determined system, in the sense that there are more column vectors than the actual rank of the matrix. This is the case for a noise-free signal series. The dimension of the column vector space is of concern and a method must be applied to reduce the rank of this space. This is the reason why the SVD of $\mathbf{XX}'$ is used.

Theorem 2.4

Suppose that a $\tau \times n$ matrix $\mathbf{X}$ has SVD

$$\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^{1/2}\mathbf{U}', \tag{2.93}$$

where

$$\begin{aligned}
\underset{(\tau \times r)}{\mathbf{V}} &= \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_r \end{bmatrix} \\
\underset{(n \times r)}{\mathbf{U}} &= \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \end{bmatrix} \\
\underset{(r \times r)}{\mathbf{\Lambda}^{1/2}} &= diag\left(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_r}\right) \\
\mathbf{V}'\mathbf{V} &= \mathbf{I}_r \\
\mathbf{U}'\mathbf{U} &= \mathbf{I}_r.
\end{aligned} \tag{2.94}$$

The column vectors of matrix $\mathbf{U}$ are the eigenvectors of the row space of matrix $\mathbf{X}$, and the column vectors of matrix $\mathbf{V}$ are the eigenvectors of the column space of matrix $\mathbf{X}$, and matrix $\mathbf{\Lambda}^{1/2}$ is a diagonal matrix consisting of the eigenvalues of matrix $\mathbf{X}$.

Matrix $\mathbf{X}\mathbf{X}'$ then has the SVD

$$\mathbf{X}\mathbf{X}' = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'. \tag{2.95}$$

Proof.

It follows by direct multiplication from (2.93) and (2.94). □

The definitions of a projection matrix, L_p-norm, Frobenius matrix norm and a scree-diagram are supplied below. These definitions will be required when dealing with optimality of SVD in the presence of noise.

Definition 2.8 Projection matrix

The projection matrix of r-flat H_r, spanned by the orthonormal set of r base vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r$ is defined by

$$\underset{(\tau \times \tau)}{\mathbf{P}_{H_r}} = \mathbf{V}\mathbf{V}', \tag{2.96}$$

where

$$\underset{(\tau \times r)}{\mathbf{V}} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_r \end{bmatrix}.$$

Definition 2.9 L_p-norm

The Euclidean L_p-norm of a matrix is defined by

$$\|\mathbf{X}\|_p = \sqrt[p]{\sum_{j=1}^{n}\sum_{i=1}^{\tau}\left|x_{ij}\right|^p}, \tag{2.97}$$

where $\left|x_{ij}\right|$ is the absolute value of the $(i,j)^{th}$ element of matrix $\mathbf{X}$. The absolute value actually measures distance, in the Euclidean sense, from the $(i,j)^{th}$ element of matrix $\mathbf{X}$ to the origin.

Definition 2.10 Frobenius Matrix Norm

The *Frobenius matrix norm* of a $\tau \times n$ matrix $\mathbf{X}$ is defined by

$$\|\mathbf{X}\|_F = \sqrt{\sum_{j=1}^{n}\sum_{i=1}^{\tau}\left|x_{ij}\right|^2} = \|\mathbf{X}\|_2. \tag{2.98}$$

The Frobenius matrix norm of two $\tau \times n$ matrices $\mathbf{X}$ and $\mathbf{Y}$, is the distance between the two matrices and defined by

$$\begin{aligned}\|\mathbf{X}-\mathbf{Y}\|_F &= \left[\sum_{j=1}^{n}\sum_{i=1}^{\tau}\left(x_{ij}-y_{ij}\right)^2\right]^{\frac{1}{2}} \\ &= \left[\sum_{j=1}^{n}\left\langle \mathbf{x}_j-\mathbf{y}_j,\mathbf{x}_j-\mathbf{y}_j\right\rangle\right]^{\frac{1}{2}}.\end{aligned} \tag{2.99}$$

Definition 2.11 Scree-diagram

A scree-diagram is a graphical plot of the coordinate pairs $\left\{\left(i,\ \lambda_{(\tau-i+1)}\right)\middle| i=1,\ldots,r\right\}$. It is therefore a plot of ordered eigenvalues, derived through SVD, against their reverse indices.

2.7.2 The effect of centring in SVD

Centring involves the process of subtracting the vector of means $(\bar{\mathbf{x}})$ from the trajectory matrix $(\mathbf{X})$ column vectors, thereby constructing the so-called centred trajectory matrix $(\tilde{\mathbf{X}})$. Geometrically, centring causes a multivariate scatter of points to be centred at the origin. In Golyandina *et al.* (2001) the comment is made that "*Centring is not a standard procedure in SVD.*" The practice of centring is central to multivariate statistics, as it forms part of the process of constructing a covariance matrix.

The trajectory matrix can be thought of as having the form

$$\begin{aligned}\mathbf{X} &= \bar{\mathbf{x}}\mathbf{1}' + \left(\mathbf{X}-\bar{\mathbf{x}}\mathbf{1}'\right) \\ &= \bar{\mathbf{x}}\mathbf{1}' + \tilde{\mathbf{X}} \\ &= \left[\bar{\mathbf{x}} \quad \bar{\mathbf{x}} \quad \cdots \quad \bar{\mathbf{x}}\right] + \left[\mathbf{x}_1-\bar{\mathbf{x}} \quad \mathbf{x}_2-\bar{\mathbf{x}} \quad \cdots \quad \mathbf{x}_n-\bar{\mathbf{x}}\right],\end{aligned} \tag{2.100}$$

where

$$\begin{aligned}\underset{(\tau\times 1)}{\bar{\mathbf{x}}} &= \frac{1}{n}\mathbf{X}\mathbf{1} \\ \underset{(\tau\times n)}{\tilde{\mathbf{X}}} &= \mathbf{X} - \bar{\mathbf{x}}\mathbf{1}' \\ \underset{(n\times 1)}{\mathbf{1}_n} &= \begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}'.\end{aligned}$$

The SVD of the trajectory matrix can therefore be written as

$$\begin{aligned}\mathbf{X} &= \bar{\mathbf{X}} + \tilde{\mathbf{X}} \\ &= \begin{bmatrix}\bar{\mathbf{x}} & \cdots & \bar{\mathbf{x}}\end{bmatrix} + \begin{bmatrix}\mathbf{x}_1 - \bar{\mathbf{x}} & \cdots & \mathbf{x}_n - \bar{\mathbf{x}}\end{bmatrix} \\ &= \sqrt{\lambda_0}\mathbf{v}_0\mathbf{u}_0' + \sum_{i=1}^{r-1}\sqrt{\lambda_i}\mathbf{v}_i\mathbf{u}_i' \ ,\end{aligned} \tag{2.101}$$

where $\mathbf{v}_1,\ldots,\mathbf{v}_{r-1}$ and $\mathbf{u}_1,\ldots,\mathbf{u}_{r-1}$ are left- and right eigenvectors of $\tilde{\mathbf{X}}$

$$\begin{aligned}\mathbf{v}_0 &= \bar{\mathbf{x}} / \|\bar{\mathbf{x}}\| \\ \mathbf{u}_0 &= \mathbf{1}_n / \sqrt{n}\end{aligned} \tag{2.102}$$

and the eigenvalue λ_0 is given by

$$\lambda_0 = \sqrt{n}\,\|\bar{\mathbf{x}}\|. \tag{2.103}$$

This is due to the fact that the matrix $\begin{bmatrix}\bar{\mathbf{x}} & \cdots & \bar{\mathbf{x}}\end{bmatrix}$ has n similar column vectors, i.e. the vector of means $\bar{\mathbf{x}}$, and is consequently of rank 1.

2.7.3 The effect of noise in SVD

A signal series $\{f_t\}_{t=1}^{N}$ from the broad class of functions in (2.57) has a trajectory matrix of finite rank r. The SVD of such a trajectory matrix will result in r non-zero eigenvalues $\sqrt{\lambda_1} \geq \sqrt{\lambda_2} \geq \ldots \geq \sqrt{\lambda_r} > 0$. When additive noise is introduced to the mentioned signal series, all eigenvalues derived through SVD will be non-zero.

Suppose that an observed time series $\{x_t\}_{t=1}^{N}$ is a noise-contaminated signal from the class in (2.57), such that

$$x_t = f_t + \varepsilon_t, \ t = 1,\ldots,N. \tag{2.104}$$

The noise component in (2.104) is hypothesised to have distributional form

$$\varepsilon_t \sim .\left(0, \ \sigma^2\right). \tag{2.105}$$

It is clear from (2.105) that no specific statistical distribution is assumed, only that the variance $\left(\sigma^2\right)$ of such a distribution be known and constant.

The unfolding of such a series into a trajectory matrix will result in the form

$$\begin{aligned} \mathbf{X} &= \mathbf{F} + \mathbf{E} \\ &= \begin{bmatrix} f_1 & \cdots & f_n \\ \vdots & \vdots & \vdots \\ f_\tau & \cdots & f_N \end{bmatrix} + \begin{bmatrix} \varepsilon_1 & \cdots & \varepsilon_n \\ \vdots & \vdots & \vdots \\ \varepsilon_\tau & \cdots & \varepsilon_N \end{bmatrix}, \end{aligned} \tag{2.106}$$

where

$$\begin{aligned} \mathbf{F} &= \text{Hankel structured matrix of the unfolded signal series} \\ \mathbf{E} &= \text{Hankel structured matrix of the unfolded stochastic series.} \end{aligned}$$

The noise, present in the trajectory matrix, will cause all eigenvalues to be non-zero. This implies that a scree-diagram, constructed from the scatter matrix of a white noise distorted signal series, will have a "noise-floor" which is fairly flat. Similar arguments can also be found in Elsner and Tsonis (1996, pp. 70-71).

Example 2.9

A series of the form

$$x_t = f_t + \varepsilon_t \ , \ t = 1,...,100, \tag{2.107}$$

where

$$\begin{aligned} f_t &= 10 + 2t \\ \varepsilon_t &\sim N\left(0,\ 50^2\right) \end{aligned} \tag{2.108}$$

was simulated for the purpose of this example. The series is therefore similar to that in Example 2.1. The only difference is the noise-variance, which is higher for illustrative purposes in this example. A trajectory matrix $(\tau = 10)$ was formed based on the noise-free signal, and SVD of the scatter matrix $\mathbf{XX}'$ then performed. The scree-diagram in Figure 2.13 below was then constructed.

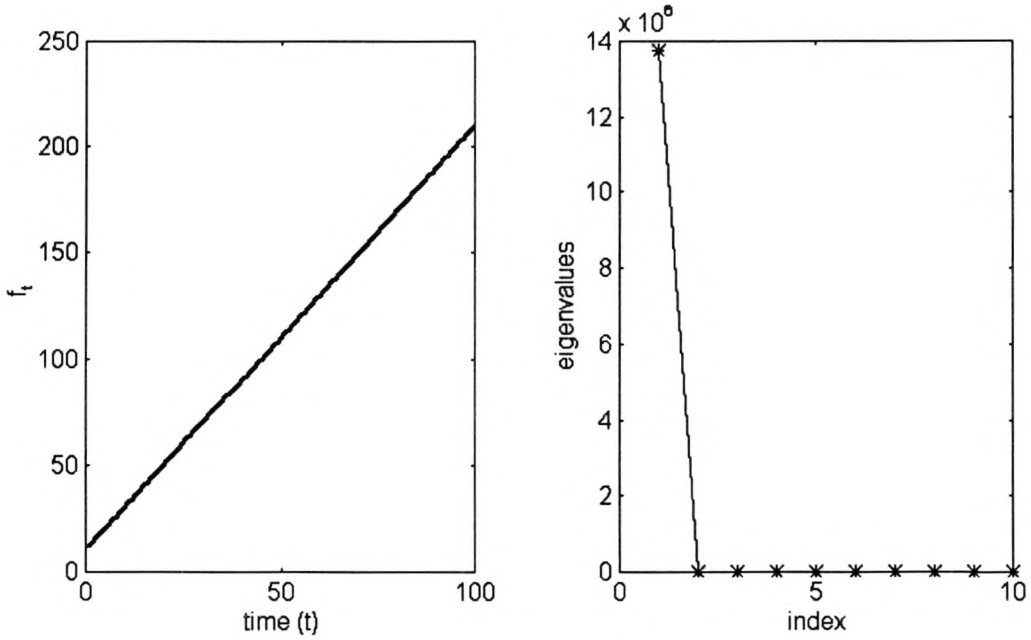


Figure 2.13 Scree-diagram of first-order polynomial signal series.

A trajectory matrix $(\tau = 10)$ was then formed based on the white noise-contaminated series, and SVD of the scatter matrix $\mathbf{XX}'$ then performed. The scree-diagram is exhibited in Figure 2.14 below. From perusal of these graphs it is clear that the noise-free signal has a single non-zero eigenvalue, cf. Figure 2.13 above.

It is clear from perusal of Figure 2.14 below that all the eigenvalues are non-zero. The first leading eigenvalue is dominant and is then followed by a fairly flat "noise floor".

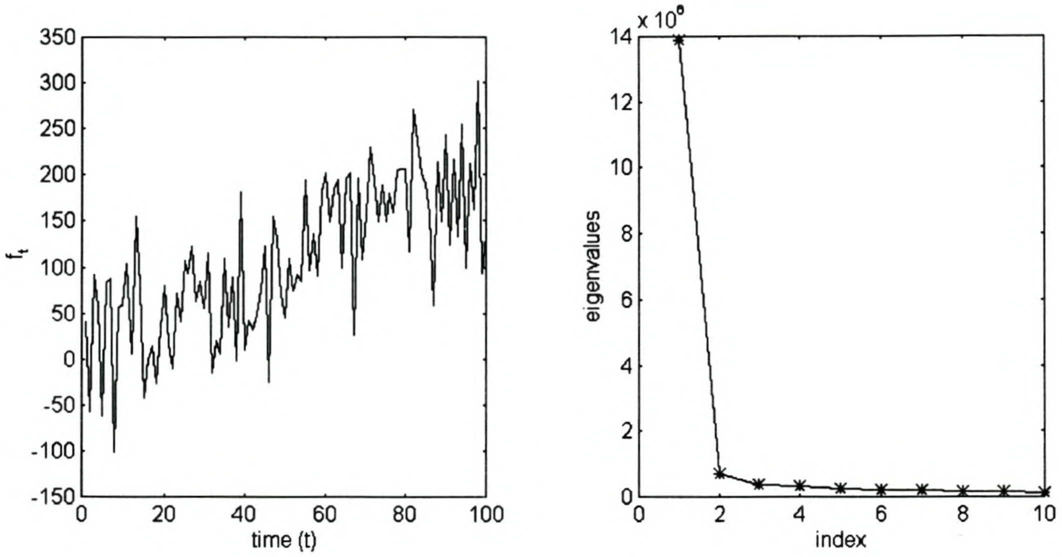


Figure 2.14 Scree-diagram of first-order polynomial white noise-contaminated series.

Example 2.10

This example will illustrate the effect of noise on the scree-diagram of a simulated sinusoidal. A series was simulated from the following class model

$$x_t = f_t + \varepsilon_t \ , \ t = 1,\ldots,100,$$

where

$$\begin{aligned} f_t &= 150\sin(2\pi t / 12 + 100) + \varepsilon_t \\ \varepsilon_t &\sim N\left(0,\ 100^2\right). \end{aligned} \tag{2.109}$$

It is clear from perusal of Figure 2.15 and Figure 2.16 below that there are two dominating leading eigenvalues.

It is once again clear that the white noise caused a relatively flat "noise-floor" in Figure 2.16 below.

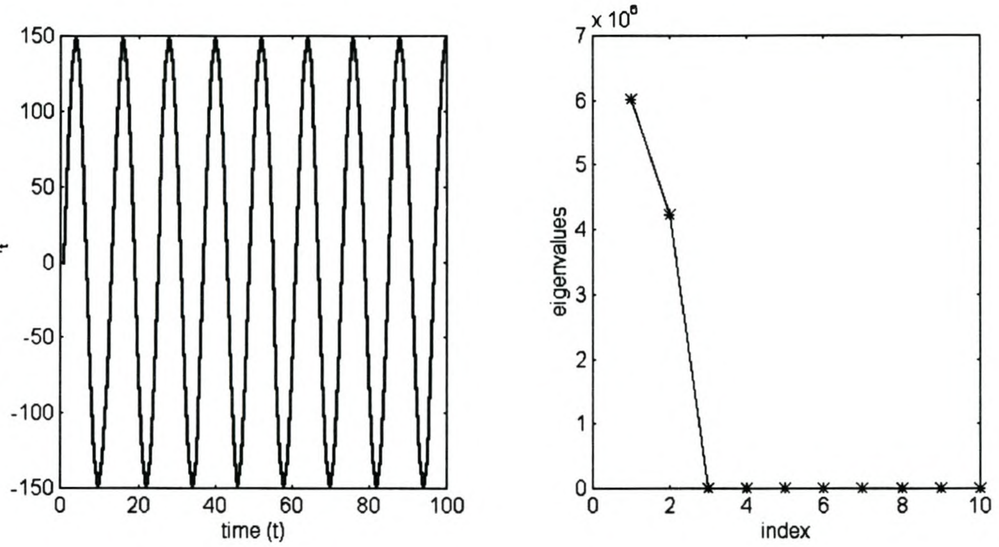


Figure 2.15 Scree-diagram of sinusoid signal series.

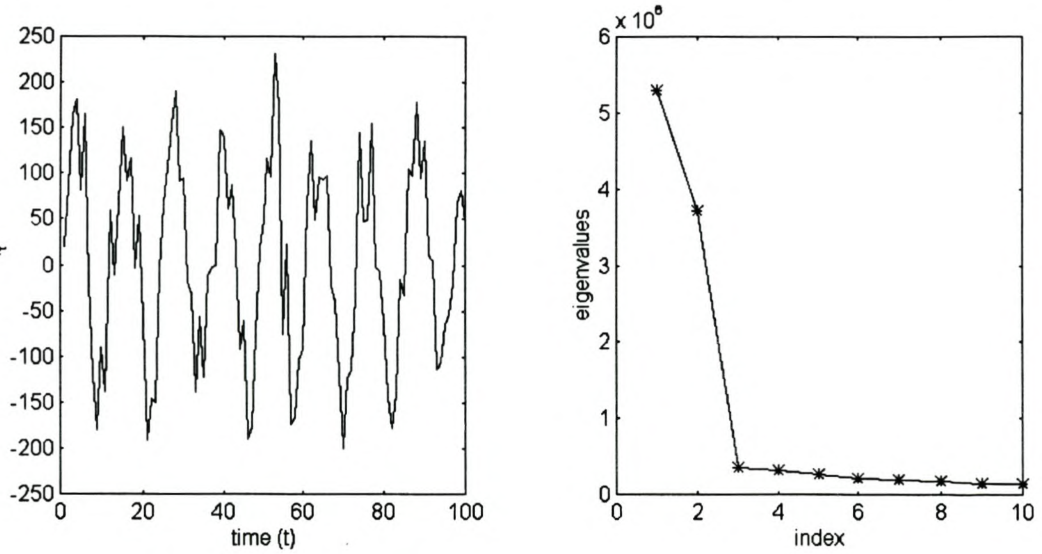


Figure 2.16 Scree-diagram of sinusoid white noise-contaminated series.

2.7.4 Optimality of SVD

The purpose of using SVD in the SSA context is to find a low-rank approximation of the trajectory matrix $(\mathbf{X})$. This issue is covered in-depth in Golyandina *et al.* (2001, Section 4.3). Arguments similar to those presented by these authors, can also be found in the spectral analysis literature (Stoica and Moses, 1997, Appendix A).

Suppose that the trajectory matrix is of rank $d \leq \min(\tau, n)$. The problem statement in SSA is to find a low-rank approximating matrix $(\mathbf{Y})$ of rank $r < d$, for the trajectory matrix $(\mathbf{X})$. This is achieved by minimising the squared Frobenius matrix norm

$$\min_{\mathbf{Y}\in\mathcal{M}_r} \|\mathbf{X} - \mathbf{Y}\|^2 , \qquad (2.110)$$

where Golyandina *et al.* (2001, p. 229) define $\mathcal{M}_r$ to "*be the set of matrices of the form*"

$$\mathbf{Y} = \sum_{i=1}^{r} \mathbf{p}_i \mathbf{q}_i' , \qquad (2.111)$$

where $\mathbf{p}_i \in \mathbb{R}^\tau$ (linearly independent) and $\mathbf{q}_i \in \mathbb{R}^\tau$ are arbitrary.

According to Golyandina *et al.* (2001, Proposition 4.5) the optimal solution to (2.110) is given by

$$\mathbf{Y}_0 = \sum_{i=1}^{r} \sqrt{\lambda_i}\,\mathbf{v}_i \mathbf{u}_i' \in \mathcal{M}_r , \qquad (2.112)$$

where $(\sqrt{\lambda_i}, \mathbf{v}_i, \mathbf{u}_i)$ are the leading eigentriples obtained through SVD of the trajectory matrix. Note that the r leading eigenvectors $\mathbf{v}_1,\ldots,\mathbf{v}_r$ can be obtained from the SVD of the scatter matrix $\mathbf{XX}'$, and that $\mathbf{u}_i = (1/\sqrt{\lambda_i})\mathbf{v}_i$

One major drawback of the "optimal result" in (2.112), is that it is not structure preserving by nature. Matrix $\mathbf{Y}_0$ does not have a Hankel structure, the key reason why SSA works. This problem seems to be receiving attention in the form of Structured Total Least Norm optimisation. A method is presented by Park *et al.* (1999) to perform such a low-rank approximation of a Hankel structured matrix, also involving use of SVD. It would, however, seem that further research is still required on this topic.

Venter (1998) viewed the dimension reduction problem in SSA, as a problem of finding the closest r-flat $\left(H_r = \mathbf{b} + span\left(\mathbf{v}_1,\ldots,\mathbf{v}_r\right)\right)$ of dimension r to the column vectors of the trajectory matrix $(\mathbf{X})$. A minimum squared distances approach was followed. The closest r-flat, of rank r, was shown to be of the form

$$H_r = \overline{\mathbf{x}} + span\left(\mathbf{v}_1,\ldots,\mathbf{v}_r\right) \tag{2.113}$$

where

$$\overline{\mathbf{x}} = \frac{1}{n}\mathbf{X}\mathbf{1}_n. \tag{2.114}$$

The set of orthonormal base vectors $\mathbf{v}_1,\ldots,\mathbf{v}_r$ are obtained through SVD of the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \left(\mathbf{X} - \overline{\mathbf{x}}\mathbf{1}_n'\right)\left(\mathbf{X} - \overline{\mathbf{x}}\mathbf{1}_n'\right)'$. Incorporating all these features, the closest rank r matrix to the trajectory matrix is given by

$$\mathbf{Y} = \overline{\mathbf{x}}\mathbf{1}' + \mathbf{P}_{span\left(\mathbf{v}_1,\ldots,\mathbf{v}_r\right)}\left(\mathbf{X} - \overline{\mathbf{x}}\mathbf{1}'\right), \tag{2.115}$$

where $\mathbf{P}_{span\left(\mathbf{v}_1,\ldots,\mathbf{v}_r\right)}$ is the projection matrix of the parallel linear subspace spanned by the r leading eigenvectors $\mathbf{v}_1,\ldots,\mathbf{v}_r$.

The result in (2.115) is also well known in the theory of biplots and resides under the heading "orthogonal projection with an offset", cf. Gower and Hand (1996, p. 254).

Example 2.11

The purpose of this example is to illustrate the use of SVD in the context of SSA. In this example a series was generated from

$$x_t = f_t + \varepsilon_t \ , \ t = 1,\ldots,100, \tag{2.116}$$

where

$$\begin{aligned} f_t &= 10 + 2t \\ \varepsilon_t &\sim N\left(0, \ 10^2\right). \end{aligned} \tag{2.117}$$

The generated series from (2.116) was unfolded into the column vectors of a trajectory matrix, using window length $\tau = 2$. The scatter plot in Figure 2.17 below exhibits the column vectors of the trajectory matrix, based on the white noise-contaminated series. The thick line represents the leading eigenvector vector, which was obtained through SVD of scatter matrix $\mathbf{XX}'$. No centring was used. The thin line represents the unfolding of the noise-free signal into a trajectory matrix. It is clear that these lines are close, but not an exact match. The greater the noise variance of an observed series, the greater this deviation will be, since SVD is not expected to result in an exact fit situation.

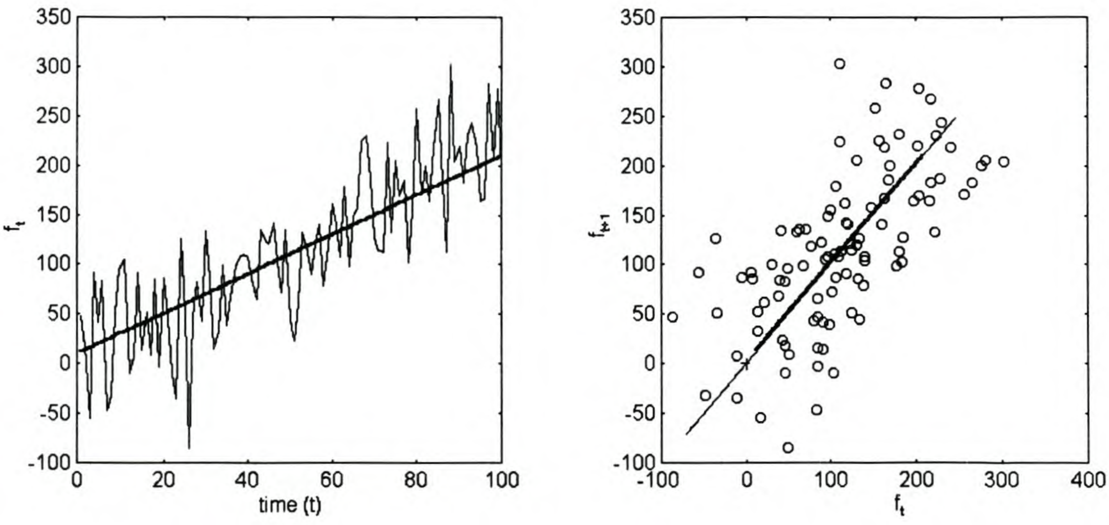


Figure 2.17 Leading eigenvector of a first-order polynomial series.

Example 2.12

In this example the extraction of eigenvectors through SVD of a trajectory matrix will once again be illustrated. A series was generated from

$$x_t = f_t + \varepsilon_t \ , \ t = 1,...,100, \tag{2.118}$$

where

$$\begin{aligned} f_t &= 150\sin(2\pi t/12 + 100) \\ \varepsilon_t &\sim N\left(0,\ 100^2\right). \end{aligned} \tag{2.119}$$

The generated series in (2.118) was unfolded into the column vectors of a trajectory matrix. No centring was used and SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}'\right)$ was performed. The leading eigenvectors are shown in Figure 2.18 below. It is clear that the thickest line represents the eigenvector that directed itself in the direction of maximum variation, in accordance with SVD theory. The second thickest line represents the second eigenvector, which directed itself in the direction of second highest variation. Note that these two eigenvectors are mutually orthogonal. The same argument holds for the last of the three eigenvectors. It is clear that the eigenvectors can be considered as a rotation of the axial system in $\mathbb{R}^3$.

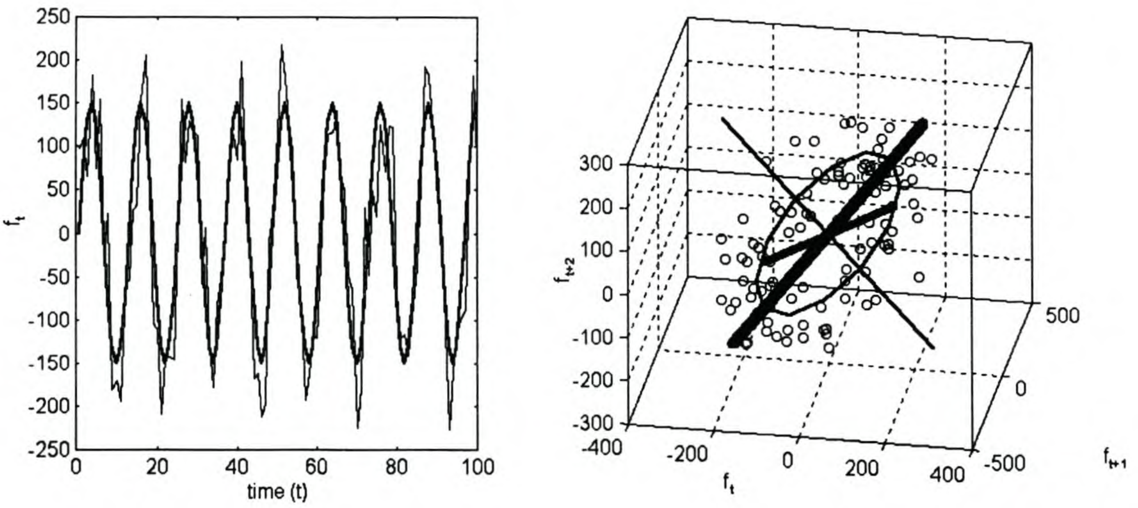


Figure 2.18 Leading eigenvectors of a sinusoidal series.

2.7.5 Artificial Phase Portraits

It was clear in §2.4.2 that the normalised eigenvectors of a trajectory matrix, based on a sinusoidal, was of the form

$$\begin{aligned}\mathbf{v}_1 &= \mathbf{y}_1 / \|\mathbf{y}_1\| \\ \mathbf{v}_2 &= \mathbf{y}_2 / \|\mathbf{y}_2\|,\end{aligned} \tag{2.120}$$

where

$$\begin{aligned}\mathbf{y}_1 &= \left[\cos(\varpi t) \quad \cos(\varpi t + \Delta t\varpi) \quad \cdots \quad \cos[\varpi t + (\tau - 1)\Delta t\varpi]\right]' \\ \mathbf{y}_2 &= \left[\sin(\varpi t) \quad \sin(\varpi t + \Delta t\varpi) \quad \cdots \quad \sin[\varpi t + (\tau - 1)\Delta t\varpi]\right]'.\end{aligned} \tag{2.121}$$

If a plot of the coordinate pairs $\{(v_{i1}, v_{i2}) | i = 1, \ldots, \tau\}$ is constructed, based on eigenvectors in (2.120), one of the following patterns will emerge.

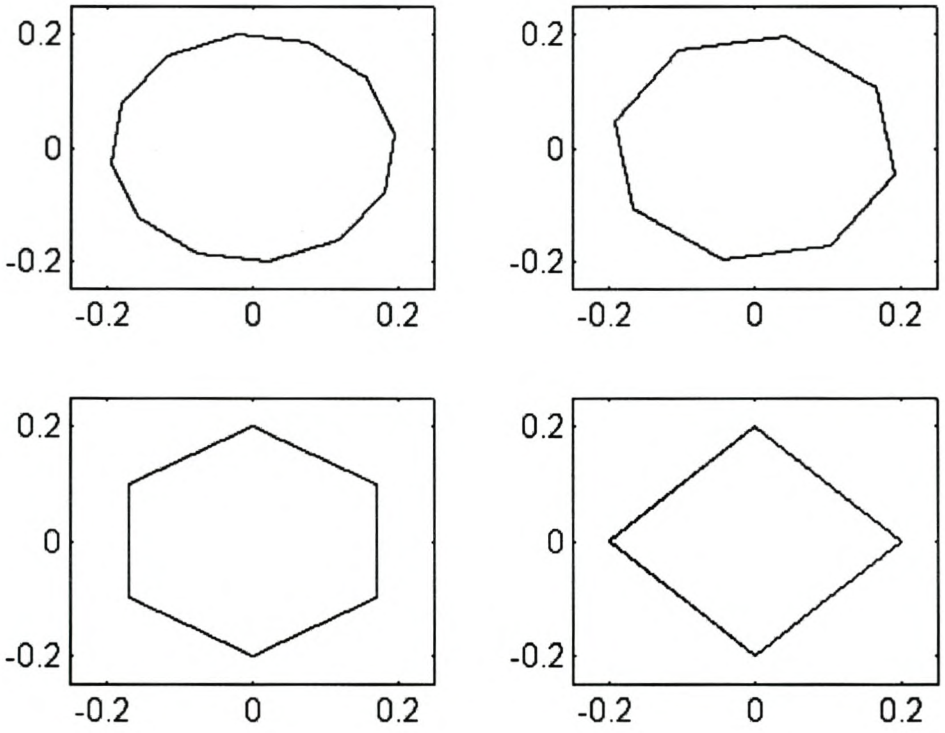


Figure 2.19 Artificial phase portraits of noise-free sinusoids.

The plot in Figure 2.19 above will be referred to as an *artificial phase portrait*. The panels (left to right) contain the artificial phase portrait of sinusoids that have periodicities $12, 8, 6$ and 4 respectively. The number of sides in each of the polygons in Figure 2.19, above, coincides with the period of the sinusoid. The signals that were used to construct the artificial phase portraits, did not contain noise. Figure 2.20 below exhibits the artificial phase portraits of noise-contaminated sinusoids.

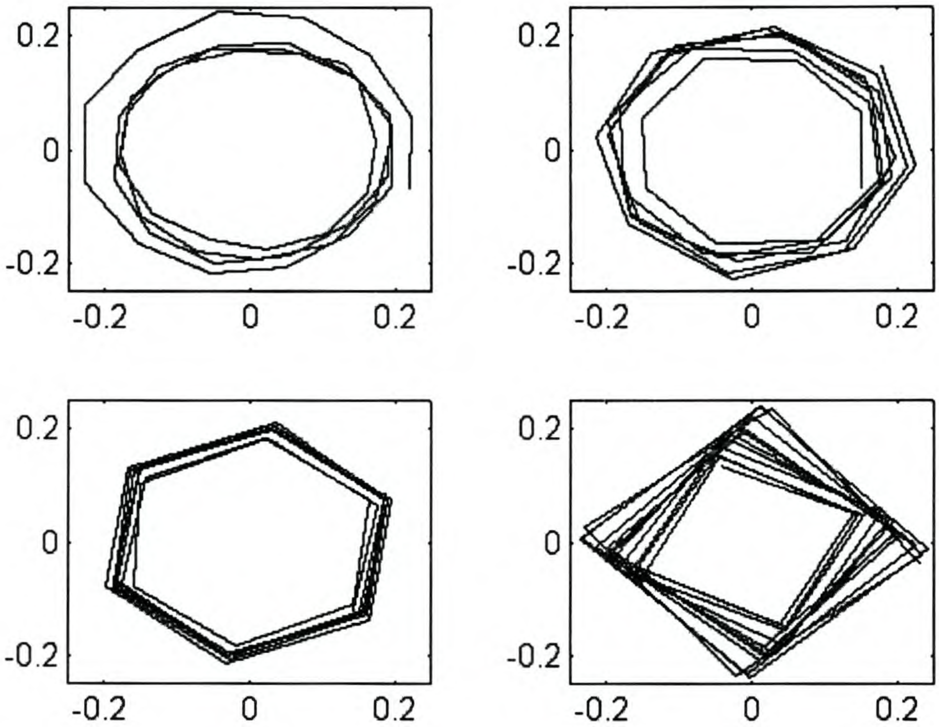


Figure 2.20 Artificial phase portraits of noise contaminated sinusoids.

It is clear that the artificial phase portrait can be used to gauge whether a sinusoid pattern is present in a time series. In an economic time series this would entail a seasonal pattern with either monthly or quarterly seasonal variation. If any other cyclical pattern is present, then the artificial phase portrait will also capture such a pattern.

The artificial phase portrait plot will be a valuable tool to aid in identifying cyclical patterns. Vautard *et al*. (1992) illustrated that SSA cannot resolve a cycle that has period longer than the window length (τ) used.

Example 2.13

The well-known airline time series, series G in Box and Jenkins (1970, p. 531), was unfolded into a time delay matrix. A window length of $\tau = 72$ was used to unfold the series. No centring of the trajectory matrix was used, and the scatter matrix $\mathbf{XX}'$ was formed. The SVD of the scatter matrix yielded 72 eigenvectors. Figure 2.21, below, exhibits a number of artificial phase space portraits, based on the eigenvectors.

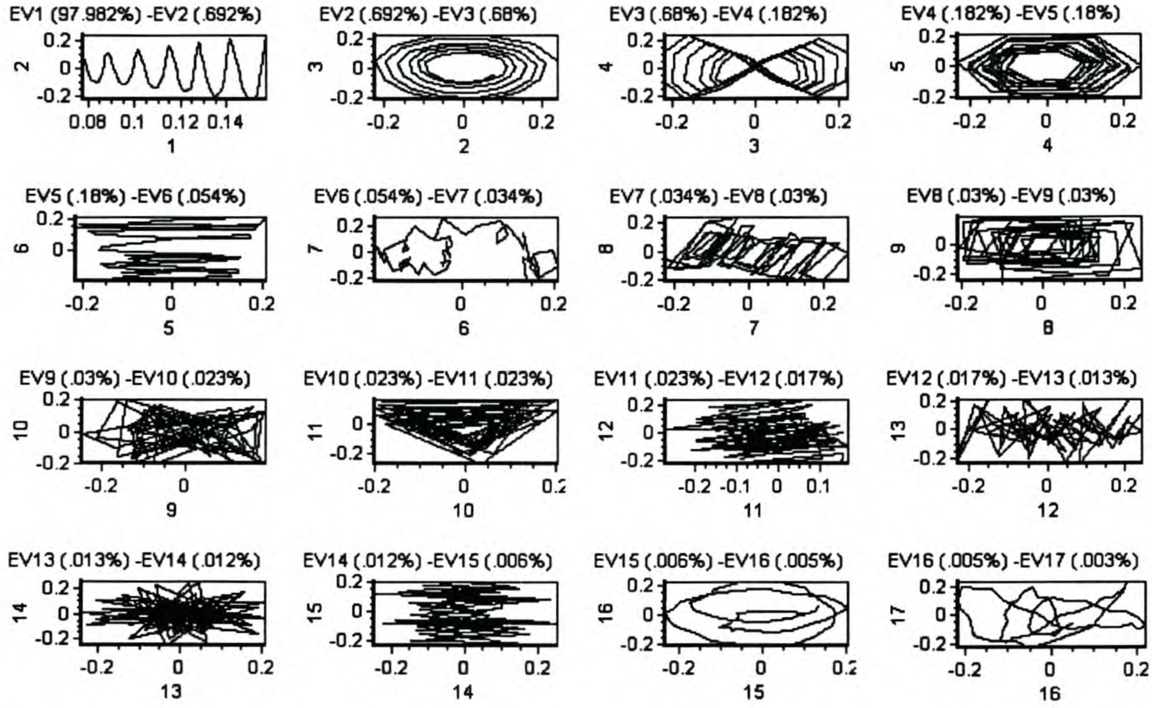


Figure 2.21 Artificial phase portraits of airline series.

It is clear from perusal of Figure 2.21 above that the following pairings of eigenvectors and associated sinusoid periodicities seems feasible, cf. Figure 2.19.

Table 2.1 Pairing of eigenvectors of airline series.

Eigenvector Pairing	Sinusoid Period
{2, 3}	12
{4, 5}	6
{8, 9}	4
{10, 11}	3

An interesting feature of the pairings in Table 2.1, above, is the percentage variation explained by the eigenvectors in a pairing. In the pairing {2,3}, the eigenvectors have respective percentage variation explained as 0.692% and 0.68%. The pairing {4,5}, have eigenvectors with respective percentage variation explained as 0.182% and 0.180%. It would therefore seem that adjacent leading eigenvectors that have fairly similar percentage variation explained, should be considered as pairings. Adjacent eigenvalues, ordered in magnitude, with fairly similar values $\left(\lambda_k \approx \lambda_{k+1}\right)$ can also be used for this purpose (Ghil and Vautard, 1991).

Another series that is known to exhibit cyclical patterns is the Wölfer sunspot series.

Example 2.14

The well-known Wölfer sunspot series is considered in this example. The series used is of length $N = 176$ and the window length set at $\tau = 88$.

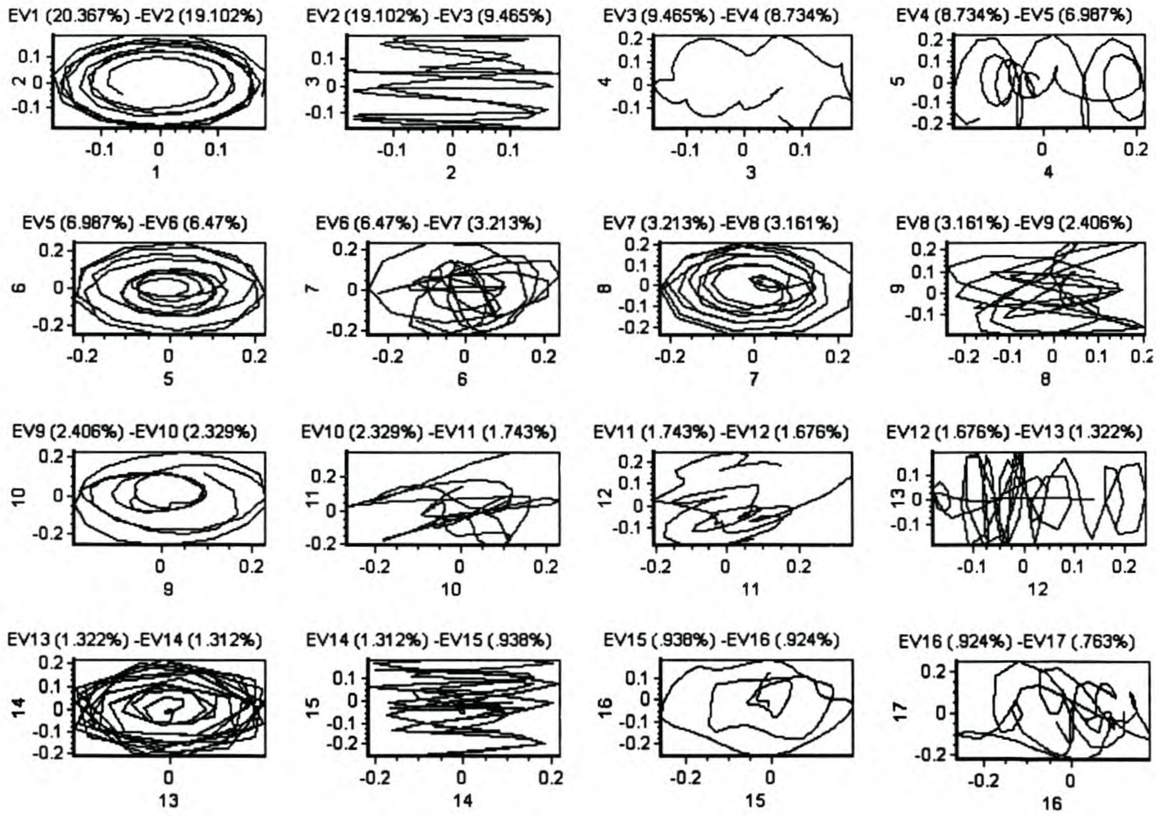


Figure 2.22 Artificial phase portraits of Wölfer sunspot series.

It would seem that a few cyclical patterns can be deduced from perusal of Figure 2.22 above. Identification of possible cycles is left to the reader's imagination.

This example is the final of this section on SVD in SSA. The foundations that were laid will now be utilised to consider signal reconstruction in the context of SSA.

2.8 Signal Reconstruction

Signal reconstruction has received little attention in the context of SSA (Golyandina *et al.*, 2001; Varadi *et al.*, 1999; Vautard *et al.*, 1992). The extraction of certain signal components from a series, such as trend and cycles, are studied in greater detail in Golyandina *et al.* (2001). In the latter case the reconstruction is considered under the topic of separability properties of a function.

Signal reconstruction is central to the field of signal processing. This section will propose the use of a method developed by Cadzow (1988) in the field of signal processing, as a signal reconstruction method in the context of SSA. No reference to the signal reconstruction method of Cadzow (1988) could be found in any of the work by Allen and Smith (1996), Buchstaber (1994), Elsner and Tsonis (1996), Fowler and Kember (1998), Golyandina *et al.* (2001), Paluš and Novotná (1998), Plaut and Vautard (1994), Vautard and Ghil (1991), Varadi *et al.* (1999) and Vautard *et al.* (1992). The method is also adapted to handle the multi-channel SSA case in §2.9 below.

2.8.1 The Hankelization Operation

Before signal reconstruction can be investigated, certain important concepts must be understood. The first concept is the Hankelization operation.

The Hankelization operation was first introduced by Buchstaber (1994), and is denoted by $\mathcal{H}$. The operation is a method that produces a Hankel structured matrix from another matrix, through a linear operation of reverse diagonal averaging. The Hankelization operation is formally defined in Definition 2.12 below.

Definition 2.12 Hankelization Operation

The Hankelization operation, performed on a $\tau \times n$ matrix $\mathbf{Z}$, creates a Hankel structured matrix $\tilde{\mathbf{Z}}$. This matrix is element-wise created as follows,

$$\tilde{z}_{ij} = \begin{cases} \dfrac{1}{i+j-1} \displaystyle\sum_{l=1}^{i+j-1} z_{l,i+j-l} & for \quad 2 \le i+j \le \tau - 1, \\ \dfrac{1}{\tau} \displaystyle\sum_{l=1}^{\tau} z_{l,i+j-l} & for \quad \tau \le i+j \le n+1, \\ \dfrac{1}{N-i-j+2} \displaystyle\sum_{l=i+j-n}^{\tau} z_{l,i+j-l} & for \quad n+2 \le i+j \le N, \end{cases} \tag{2.122}$$

for $i = 1,...,\tau$ and $j = 1,...,n$. The notation $\tilde{\mathbf{Z}} = \mathcal{H}\mathbf{Z}$ will be used to denote the linear operation defined in (2.122).

The Hankelization operation creates a matrix that has Hankel structure, such that the Frobenius matrix norm $\left\|\mathbf{Z} - \tilde{\mathbf{Z}}\right\|_F$ is minimal. The proof can be found in Golyandina *et al.* (2001, p. 266).

The following algorithm also performs Hankelization of a $\tau \times n$ matrix $\mathbf{Z}$, and results in a "signal series" $\left\{\tilde{f}_t\right\}_{t=1}^{N}$

$$\tilde{f}_s = \begin{cases} \dfrac{1}{s} \displaystyle\sum_{i=1}^{s} z_{i,s-i+1} & for \quad 1 \le s \le \tau, \\ \dfrac{1}{\tau} \displaystyle\sum_{i=1}^{\tau} z_{i,s-i+1} & for \quad \tau \le s \le n, \\ \dfrac{1}{N-s+1} \displaystyle\sum_{i=1}^{N-s+1} z_{i+s-n,n-i+1} & for \quad n \le s \le N. \end{cases} \tag{2.123}$$

2.8.2 The Approximate Series

Suppose now that an approximate series (a signal series) must be reconstructed from a noise-contaminated series. Alternatively, that a certain component (trend or sinusoid) of a noise-contaminated or noise-free series must be separated through a re-constructive approach. The approximate series must therefore have a trajectory matrix of fixed rank, (say) r.

The approach is provided in the form of Algorithm 2.1 below.

Algorithm 2.1 Construction of an Approximate Series

(a) Unfold an observed series $\{x_t\}_{t=1}^{N}$ into a trajectory matrix $\mathbf{X}$.

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}} = \mathbf{X} - \bar{\mathbf{x}}\mathbf{1}'$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix, where $\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'$.

(e) Select only r eigenvectors in the columns of matrix $\mathbf{V}$.

(f) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r} = \mathbf{V}\mathbf{V}'$.

(g) Form the projected result $\mathbf{Z} = \bar{\mathbf{x}}\mathbf{1}' + \mathbf{P}_{\mathfrak{L}_r}\tilde{\mathbf{X}}$.

(h) Perform the Hankelization operation $\tilde{\mathbf{Z}} = \mathcal{H}\mathbf{Z}$ using (2.122).

(i) Use (2.123) to form the reconstructed series $\{\tilde{f}_t\}_{t=1}^{N}$.

The column vectors of the Hankel structured trajectory matrix of $\{\tilde{f}_t\}_{t=1}^{N}$ will not reside on a single r-flat. The linear Hankelization matrix operation, in step (h), is performed after the projection. The projection, in step (g), ensures that column vectors of the resulting matrix $\mathbf{Z}$ reside on a single r-flat. This was disturbed by the Hanekelization operation, to maintain a Hankel structured matrix.

There is another undesired and unavoidable property that the Hankelization operation possesses. The Achilles heel of this operation can be coined the “end-effects”. It is clear from (2.123) that reverse diagonal averaging is used to produce the approximate series. On closer inspection of (2.123), it is clear that there are more observations that are averaged over in the middle section of a time series, and less towards the ends of the series. In the case of the first and last observations, only a single element is averaged over.

It is the undesired property of column vectors that do not lie on a single r-flat that initiated further research in this regard. The construction of a signal using an iterative property mapping algorithm, proposed by Cadzow (1988), will be considered next.

2.8.3 Signal Reconstruction using Cadzow's Algorithm

As mentioned, this study is the first to propose the use of a method developed by Cadzow (1988) in the context of SSA.

Cadzow (1988) proposed a composite property-mapping algorithm that reconstructs a signal series. The technique unfolds a noise-contaminated series into a Toeplitz structured trajectory matrix (see p.20) or any other linear structured matrix. Hankel structured matrices are also linear and briefly mentioned in the paper. Not many published research articles could be found, using a citation index search, that reference Cadzow's proposal (Hurt, 1991; Park *et al*, 1999). To date, no proof has been put forward to explain why Cadzow's algorithm actually converges, other than an intuitive reasoning in the paper. This still remains an open issue and further research is required in this regard.

The application of Cadzow's algorithm is diagrammatically illustrated in Figure 2.23 below. According to Cadzow's paper the algorithm is used for,

> "*...recovering a signal from a noise-contaminated distorted measurement made on that signal. The measured signal can take on such distinctive forms as being*

- *a series of discrete-time signal observations*
- *set of autocorrelation lag estimates*
- *sampled two-dimensional image*
- *excitation-response observations made in a system identification application.*"

Note that Cadzow's algorithm requires prior knowledge of the window length (τ) and number of eigenvectors (r). In signal processing, engineers often have prior information on the structure of the underlying signal. This might make it possible to supply values for both τ and r.

This thesis, however, will make the assumption that there is no *a priori* knowledge of τ and r, and that these quantities must be found using some form of model validation technique. This issue will be addressed in Chapter 4.

Cadzow's composite property mapping algorithm will now be formulated and can be followed by viewing Figure 2.23 below.

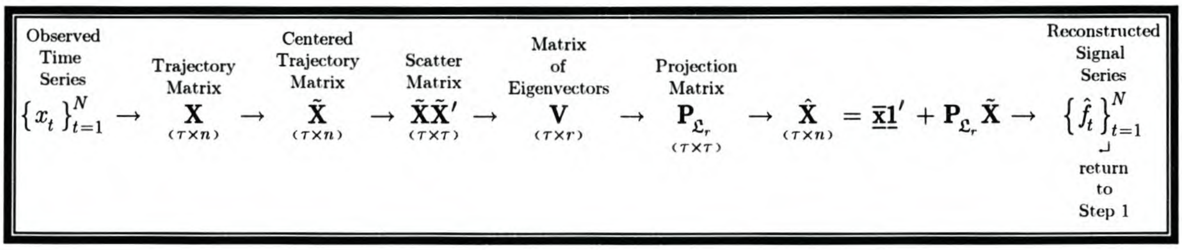


Figure 2.23 Flow-diagram of Cadzow's iterative property mapping algorithm.

Algorithm 2.2 Construction of a Cadzow-signal.

(a) Unfold an observed series $\{x_t\}_{t=1}^N$ into a trajectory matrix $\mathbf{X}$.

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}} = \mathbf{X} - \bar{\mathbf{x}}\mathbf{1}'$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix, where $\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}'$.

(e) Select only r eigenvectors in the columns of matrix $\mathbf{V}$.

(f) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r} = \mathbf{V}\mathbf{V}'$.

(g) Form the projected result $\mathbf{Z} = \bar{\mathbf{x}}\mathbf{1}' + \mathbf{P}_{\mathfrak{L}_r}\tilde{\mathbf{X}}$.

(h) Use (2.123) to form the reconstructed series $\{\tilde{f}_t\}_{t=1}^N$. This inherently performs the Hankelization operation and produces a single signal series. Test for convergence by testing for $\sum_{i=r+1}^{\tau} \lambda_i / \sum_{i=1}^{\tau} \lambda_i < \varepsilon$. Epsilon is a convergence criterion and is set arbitrarily. A value of $\varepsilon = 10^{-7}$ was used throughout this study. If the convergence criterion is not met, the series obtained in step (h) is returned as new "observed series" in step (a). The process is iterated until convergence.

Example 2.15

The effective use of Algorithm 2.2, above, is illustrated in this example. The airline series is used, due to the fact that it contains both cyclical patterns as well as trend. The series contains heteroskedasticity, and is customarily log-transformed. A log-transform will not be made use of, to illustrate the power of the non-parametric nature of SSA.

A window length of $\tau = 72$ was set, and only the leading eigenvector used to construct Figure 2.24 below.

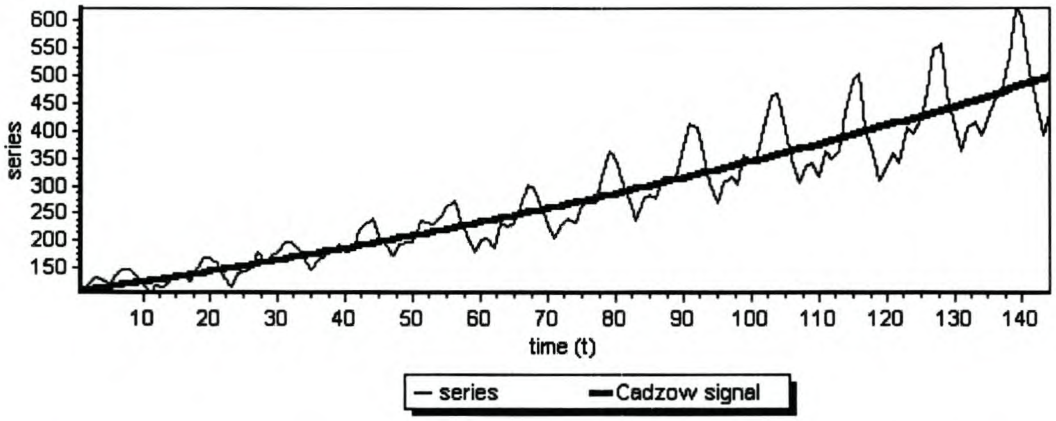


Figure 2.24 Reconstructed Cadzow-signal for airline series. $(\tau = 72,\ r = 1)$

It is clear that the leading eigenvector can be used to model trend in this particular series. This is not generally the case. It is also the first time that the power of the non-parametric nature of SSA is seen in this study.

In Figure 2.25, below, a window length $\tau = 72$ was used. The first 3 leading eigenvectors were used in Algorithm 2.2. It is clear that these three eigenvectors explain a substantial amount of variation in the airline series. The exponential trend and increasing amplitude of the sinusoid, evidently exhibits a close resemblance to the series.

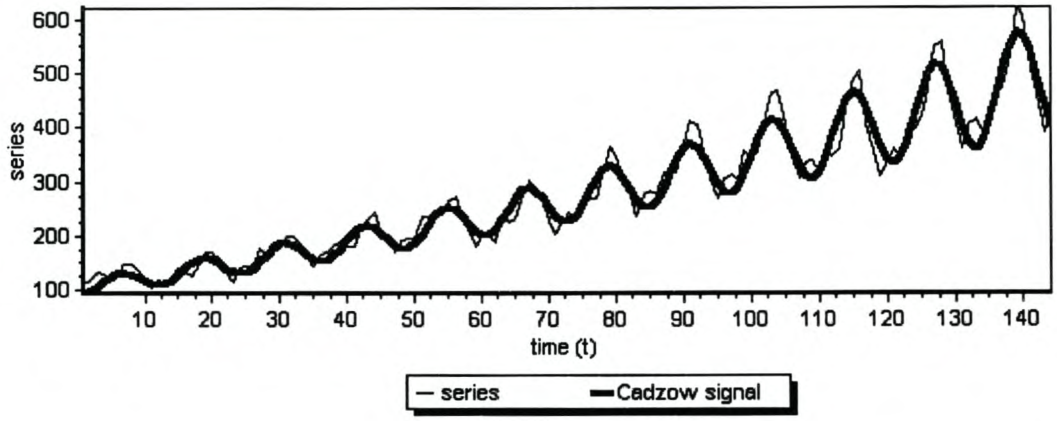


Figure 2.25 Reconstructed Cadzow-signal for airline series. $(\tau = 72,\ r = 3)$

Finally, the window length was set at $\tau = 72$ and the 5 leading eigenvectors in Algorithm 2.2 employed. The result is Figure 2.26 below. It is clear that an additional harmonic is introduced, due to the inclusion of the additional eigenvector pair $\{4,5\}$. This is in accordance with the results entertained in Example 2.13 on page 71, cf. Table 2.1.

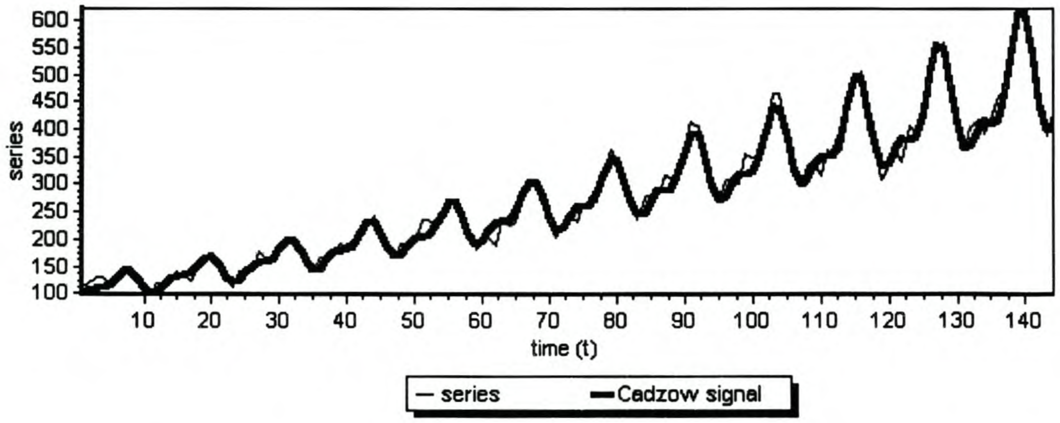


Figure 2.26 Reconstructed Cadzow-signal for airline series. $(\tau = 72,\ r = 5)$

2.8.4 Other possible signal reconstruction methods

In the SSA context, a reconstructed signal series must be such that the column vectors of its trajectory matrix reside on a single r-flat H_r.

This problem can also be dealt with on an optimisation level. An optimisation method that strives to find a low rank Hankel approximation of a matrix, is proposed by Park *et al.* (1999). It would seem that further research is still required in this regard. The proposed method was therefore not pursued in this study.

If a future optimisation method, that produces a specified low rank approximation of a Hankel structure matrix, is found, then the reconstructed signal can be found using Algorithm 2.3 below.

Algorithm 2.3 Signal reconstruction using a low rank Hankel structured matrix.

(a) Unfold an observed time series into a Hankel structured matrix $\mathbf{X}$.

(b) Specify the desired rank of the matrix to be (say) r, and perform a low rank optimisation, in the form of matrix $\mathbf{Z}$.

(c) Use (2.123) to produce the required signal series. The formulation in (2.123) will simply perform reverse diagonal averaging, and not influence the result obtained in step (b).

The signal reconstruction method, using the Cadzow (1988) algorithm, was found to be superior to the approximate series method. The algorithm also seemed to be very robust in the presence of outliers. This feature will be further explored in Chapter 4.

Section 2.9 below briefly extends the single-channel SSA ideas to the multi-channel SSA situation. The section on signal reconstruction, using Cadzow (1988), might prove to be beneficial since such a method has not been proposed to date.

2.9 Multi-Channel SSA

A brief account of multi-channel SSA can be found in (Plaut and Vautard, 1994; Elsner and Tsonis, 1996). This section will be devoted to the components that make up multi-channel SSA. The theoretical setting makes multivariate time series analysis possible.

2.9.1 The Trajectory Matrix

Suppose that p time series of common length N is available. Denote the individual time series as follows

$$\left\{f_t^{(k)}\right\}_{t=1}^{N}, \; k = 1,\dots,p, \tag{2.124}$$

where $f_t^{(k)}$ represents observation t of the k^{th} series. Each of the time series is then unfolded into a trajectory matrix, where

$$\underset{(\tau\times n)}{\mathbf{F}_k} = \begin{bmatrix} f_1^{(k)} & f_2^{(k)} & \cdots & f_n^{(k)} \\ f_2^{(k)} & f_3^{(k)} & \cdots & f_{n+1}^{(k)} \\ \vdots & \vdots & \vdots & \vdots \\ f_\tau^{(k)} & f_{\tau+1}^{(k)} & \cdots & f_N^{(k)} \end{bmatrix} \tag{2.125}$$

denotes the trajectory matrix of the k^{th} series. The individual trajectory matrices are then used to form a single trajectory matrix of the form

$$\underset{(p\tau\times n)}{\mathbf{X}} = \begin{bmatrix} f_1^{(1)} & f_2^{(1)} & \cdots & f_n^{(1)} \\ f_1^{(2)} & f_2^{(2)} & \cdots & f_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ f_1^{(p)} & f_2^{(p)} & \cdots & f_n^{(p)} \\ \vdots & \vdots & \vdots & \vdots \\ f_\tau^{(1)} & f_{\tau+1}^{(1)} & \cdots & f_N^{(1)} \\ f_\tau^{(2)} & f_{\tau+1}^{(2)} & \cdots & f_N^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ f_\tau^{(p)} & f_{\tau+1}^{(p)} & \cdots & f_N^{(p)} \end{bmatrix} = \underset{(\tau\times n)}{\mathbf{F}_1} \otimes \underset{(p\times 1)}{\mathbf{e}_1} + \underset{(\tau\times n)}{\mathbf{F}_2} \otimes \underset{(p\times 1)}{\mathbf{e}_2} + \cdots + \underset{(\tau\times n)}{\mathbf{F}_p} \otimes \underset{(p\times 1)}{\mathbf{e}_p}, \tag{2.126}$$

where $\mathbf{e}_j$ is the j^{th} column vector of the $p \times p$ identity matrix $\mathbf{I}_p$. The nature of the unfolding in (2.126), dictates that each of the p series be of common length N. It also forces the use of common window length τ. Although the trajectory matrix featuring in (2.126) seems rather complex, the following multi-channel SSA theory is analogous to that of single channel SSA.

2.9.2 The Centred Trajectory Matrix

The centred trajectory matrix is obtained by subtracting the vector of means from each column vector of the trajectory matrix in (2.126), viz.

$$\underset{(p\tau\times n)}{\tilde{\mathbf{X}}} = \mathbf{X} - \bar{\mathbf{x}}\mathbf{1}_\mathrm{n}' \tag{2.127}$$

where

$$\begin{aligned} \underset{(p\tau\times 1)}{\bar{\mathbf{x}}} &= \frac{1}{n}\mathbf{X}\mathbf{1} \\ \underset{(n\times 1)}{\mathbf{1}_n} &= \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}'. \end{aligned} \tag{2.128}$$

2.9.3 The Scatter Matrix

Depending on whether centring is applied to the trajectory matrix or not, the scatter matrix will be given by

$$\underset{(p\tau\times p\tau)}{\tilde{\mathbf{X}}\tilde{\mathbf{X}}'} = \left(\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}'\right)\left(\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}'\right)' \tag{2.129}$$

or

$$\underset{(p\tau\times p\tau)}{\mathbf{X}\mathbf{X}'}. \tag{2.130}$$

2.9.4 SVD of the Scatter Matrix

The SVD of the scatter matrix in §2.9.3 is given by

$$\underset{(p\tau\times p\tau)}{\tilde{\mathbf{X}}\tilde{\mathbf{X}}'} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}', \tag{2.131}$$

where

$$\begin{aligned} \underset{(p\tau\times p\tau)}{\mathbf{V}} &= \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_{p\tau} \end{bmatrix} \\ \mathbf{\Lambda} &= diag\left(\lambda_1,\ldots,\lambda_{p\tau}\right). \end{aligned} \tag{2.132}$$

2.9.5 The Hankelization Operation

The Hankelization operation in (2.122), on page 73, can be extended to handle the multi-channel SSA situation. Given a $p\tau \times p\tau$ matrix $\mathbf{Z}$, a Hankel structured matrix can be element-wise constructed using (2.133) below. The resulting Hankel structured matrix is denoted by $\tilde{\mathbf{Z}}$.

$$\tilde{z}_{ij}^{(k)} = \begin{cases} \dfrac{1}{i+j-1}\displaystyle\sum_{l=1}^{i+j-1} z_{(k-1)\times\tau+l,i+j-l}^{(k)} & for \quad 2 \leq i+j \leq \tau-1, \\ \dfrac{1}{\tau}\displaystyle\sum_{l=1}^{\tau} z_{(k-1)\times\tau+l,i+j-l}^{(k)} & for \quad \tau \leq i+j \leq n+1, \\ \dfrac{1}{N-i-j+2}\displaystyle\sum_{l=i+j-n}^{\tau} z_{(k-1)\times\tau+l,i+j-l}^{(k)} & for \quad n+2 \leq i+j \leq N, \end{cases} \tag{2.133}$$

If the Hankelization operation is applied in the context of a multi-channel trajectory matrix, then reverse diagonal averaging in (2.133) assumes that the trajectory matrix actually has the form

$$\underset{(p\tau\times n)}{\mathbf{X}} = \underset{(p\times1)}{\mathbf{e}_1} \otimes \underset{(\tau\times n)}{\mathbf{F}_1} + \underset{(p\times1)}{\mathbf{e}_2} \otimes \underset{(\tau\times n)}{\mathbf{F}_2} + \cdots + \underset{(p\times1)}{\mathbf{e}_p} \otimes \underset{(\tau\times n)}{\mathbf{F}_p} \ . \tag{2.134}$$

This is not the case from perusal of (2.126) above. The row vectors of a multi-channel trajectory matrix must therefore be interchanged to have the required form in (2.134). Only then, can the Hankelization operation in (2.133) be applied.

2.9.6 Reconstruction of an Approximate Series

Reconstruction of an approximate series uses the same steps as Algorithm 2.1 above. The required algorithm is supplied below.

Algorithm 2.4 Construction of an approximate series in multi-channel SSA.

(a) Unfold p observed time series into individual trajectory matrices $\{\mathbf{F}_k,\ k = 1,\ldots,p\}$ using (2.125). Form a single trajectory matrix $\mathbf{X}$, using (2.126).

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}} = \mathbf{X} - \overline{\mathbf{x}}\mathbf{1}'$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix, where $\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'$.

(e) Select only r eigenvectors in the columns of matrix $\mathbf{V}$.

(f) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r} = \mathbf{V}\mathbf{V}'$.

(g) Form the projected result $\hat{\mathbf{X}} = \overline{\mathbf{x}}\mathbf{1}' + \mathbf{P}_{\mathfrak{L}_r}\tilde{\mathbf{X}}$.

(h) Interchange the column vectors of the projected result in (g) to have the required form in (2.134). Construct the p approximate series, using the Hankelization operation $\tilde{\mathbf{Z}} = \mathcal{H}\mathbf{Z}$.

2.9.7 Reconstructing a Signal using Cadzow's Algorithm

Algorithm 2.5 below illustrates how signal series can be reconstructed in the multi-channel scenario, by adapting the Cadzow (1988) algorithm for this purpose.

Algorithm 2.5 Construction of a Cadzow-signal in multi-channel SSA.

(a) Unfold p observed time series into individual trajectory matrices $\{\mathbf{F}_k,\ k = 1,\ldots,p\}$ using (2.125). Form a single trajectory matrix $\mathbf{X}$, using (2.126).

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}} = \mathbf{X} - \overline{\mathbf{x}}\mathbf{1}'$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix, where $\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V\Lambda V}'$.

(e) Select only the r leading eigenvectors in the columns of matrix $\mathbf{V}$.

(f) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r} = \mathbf{VV}'$.

(g) Form the projected result $\mathbf{Z} = \overline{\mathbf{x}}\mathbf{1}' + \mathbf{P}_{\mathfrak{L}_r}\tilde{\mathbf{X}}$.

(h) Interchange the column vectors of the projected result in (g) to have the required form in (2.134). Perform the Hankelization operation $\tilde{\mathbf{Z}} = \mathcal{H}\mathbf{Z}$.

(i) Form the reconstructed series $\{\tilde{f}_t\}_{t=1}^{N}$. This inherently performs the Hankelization operation and produces the p signal series. Test for convergence by testing for $\sum_{i=r+1}^{T}\lambda_i / \sum_{i=1}^{T}\lambda_i < \varepsilon$. Epsilon is a convergence criteria and is set arbitrarily. A value of $\varepsilon = 1e-7$ was used throughout this study. If the convergence criteria is not met, the p signal series obtained in step (i) are returned as new "observed series" in step (a). The process is iterated until convergence.

2.10 Conclusions and Recommendations

This Chapter considered the elements that make up SSA and traced back the roots of SSA to the field of physics. SSA unfolds an observed time series into a trajectory matrix, which has a Hankel structure. Section 2.3 outlined the theoretical considerations that govern the trajectory matrix.

The broad class of functions that SSA can handle was considered in Section 2.4. A broad class of functions that has an additive property was considered in that section. Insight into this class of functions is due to the groundbreaking article of Buchstaber (1994). SSA can essentially handle functions that are governed by a linear recurrent formula and includes the broad class of functions that was proposed by Buchstaber (1994). Section 2.5, briefly, introduced the notion of linear recurrent formula. Golyandina *et al.* (2001) can be consulted for a more complete treatment of the latter viewpoint.

Section 2.6 considered SSA models that contain white noise and noise with a first-order autoregressive structure. Multivariate distributional properties were also derived under the latter assumptions. The derivations in this section are required in Chapter 4, where robust forms of model selection procedures are considered in Section 4.4.6. Difficulties that arise in many of the available statistical robustification procedures (PCA and SVD) are considered in that section, and are explained with Section 2.6 as backdrop.

Singular value decomposition is a well-known dimension reduction technique that is used in numerical linear algebraic- and in statistical applications. Section 2.7 was devoted to the use of SVD in SSA.

Cadzow's (1988) iterative property mapping algorithm was used to construct a Cadzow-signal in Section 2.8. The section dealt with signal reconstruction in SSA, in general. This thesis is probably one of the first studies to use the Cadzow-signal series. Golyandina *et al.* (2001) proposed the use of an approximate series, which is actually a signal series resulting from a single iteration in the Cadzow-signal reconstruction algorithm. The section outlined the need for signal reconstruction and also pointed out the Achilles heel of the currently available signal reconstruction methods. This was termed the "end-effects" (cf. Section

2.8.2). It was pointed out that further research is required in that regard. A method must be developed that constructs a pre-specified, low-rank approximation of a Hankel structured matrix. It was proposed that the STLN (Park *et al.*, 1999) approach be considered in this regard.

Section 2.9 considered multi-channel SSA. Formulae were provided that were used to construct the trajectory matrix, shift vector, scatter matrix and perform SVD. The section also provided algorithms that can be used to perform signal reconstruction in multi-channel SSA. Available literature on multi-channel SSA is relatively sparse, which might be an indication of relatively uncharted territory…

Chapter 3

FORECASTING IN SSA

3.1 Introduction

Chapter 2 introduced the basic underlying elements in SSA. The most important result in Chapter 2 was the fact that SSA can handle functions, which are governed by LRF. A broad class of functions (Buchstaber, 1994), for which this is true, was considered.

This Chapter considers forecasting in SSA and builds on the foundations that were laid in Chapter 2. Various algorithmic approaches to forecasting, in SSA, are considered and theoretical justifications provided. The first reference on forecasting in SSA dates back to Zhang *et al.* (1993). The latter forecasting approach was iterative by nature. The technique became outdated when the first algorithmic approach to recurrent one-period-ahead forecasting was developed (Danilov, 1997). The latter advance is a direct result of the groundbreaking article of Buchstaber (1994). Algorithm 3.1 in Section 3.2.1 is an algorithmic approach based on Danilov's (1997) proposal, and can be used to produce one-period-ahead forecasts. Recurrent application of the latter algorithm can be used to produce k -period-ahead forecasts.

Venter (1998) proposed a different, coordinate-free, approach to recurrent one-period-ahead forecasting. Algorithm 3.2 in Section 3.2.2 gives an algorithmic approach for the latter method. Recurrent one-period-ahead forecasting is also discussed in Golyandina *et al.* (2001). Their proposal is contained in Algorithm 3.4 in Section 3.2.4. Their recurrent one-period-ahead forecasting proposal uses the notion of linear recurrent formulae, which leads to a different set of forecasting formulae. The basic underlying theory is, however, the same for all the one-period-ahead forecasting algorithms considered in this chapter. Section 3.2.5 provides a proof that the recurrent one-period-ahead forecasting algorithm of

Golyandina *et al.* (2001) and the coordinate-free forecasting algorithm of Venter (1998) are equivalent.

This thesis identified the need for a forecasting algorithm that could produce joint-horizon k-period-ahead forecasts directly, instead of using a one-period-ahead forecasting algorithm recurrently. An alternative one-period-ahead formulation was developed and is considered in Section 3.2.3. The proposed one-period-ahead forecasting algorithm is generalized in Section 3.4.1 to produce joint-horizon k-period-ahead forecasts. Algorithm 3.10 is the main contribution of this Chapter. This algorithm can produce joint horizon k-period-ahead forecasts directly. Recurrent one-period-ahead forecasts can, as a special case, also be produced by this algorithm. It should be clear that this algorithm is indeed very general and the formulation can replace all the recurrent one-period-ahead formulations due to Danilov (1994), Venter (1998) and Golyandina *et al.* (2001).

Section 3.2 mainly considers recurrent one-period-ahead forecasting, when noise-free signal series are available. This assumption is relaxed in Section 3.3, when the algorithms are adapted to handle noise-contaminated series. Section 3.3.1 adapts all the recurrent one-period-ahead forecasting algorithms, to use an approximate series. The use of an approximate series is a proposal due to Golyandina *et al.* (2001). Section 3.3.2 adapts all the recurrent one-period-ahead forecasting algorithms, to use a reconstructed Cadzow-signal series. A literature study revealed that the iterative property mapping algorithm of Cadzow (1988) has never been used before in SSA.

Section 3.4 considers the construction of k-period-ahead forecasts. The main contribution of this section is the joint-horizon k-period-ahead forecasting algorithm in the form of Algorithm 3.10. The section also considers the k-period-ahead vector forecasting approach of Golyandina *et al.* (2001), cf. Algorithm 3.11.

Section 3.5 relaxes the assumption of an observed noise-free signal, and considers k-period-ahead forecasting in the presence of additive white noise.

The proposed joint-horizon k-period-ahead forecasting algorithm (cf. Algorithm 3.10) is adapted in Section 3.6, to produce forecasts in the multi-channel SSA case. This is also a contribution to SSA forecasting, as no similar forecasting method was available at the time of writing this thesis.

Issues relating to forecasting accuracy, in the SSA context, are dealt with in Section 3.7. The verticality condition, separability of functions, "end-effects" of the Hankelization operation, presence of outliers and the signal-to-noise ratio are considered in detail.

A number of examples are presented in Section 3.8 and the Chapter is concluded in Section 3.9.

3.2 Recurrent one-period-ahead forecasting algorithms for noise-free signals

This section introduces a number of algorithms that have been proposed to handle recurrent one-period-ahead forecasting (Danilov, 1997; Venter 1998, Golyandina *et al.*, 2001).

In this Chapter, the assumption is made that an observed time series $\{f_t\}_{t=1}^{N}$ is governed by a linear recurrent formula (LRF). This assumption is thoroughly discussed in Golyandina *et al.* (2001). Functions that are governed by LRF include the following broad class

$$f(t) = \sum_{k=1}^{K} p_k(t)\exp(\lambda_k t)\sin(\varpi_k t + \varphi_k). \tag{3.1}$$

This class was introduced by Buchstaber (1994), and thoroughly discussed in Chapter 2. The underlying SSA theory remains the same irrespective of whether the class of functions in (3.1) or the notion of an LRF is used. When the broad class of functions in (3.1) is therefore mentioned it should be equivalently taken as considering any function governed by an LRF and vice versa.

The assumption of an observed noise-free series will be relaxed in Section 3.3, where noise-contaminated series will be considered.

A recurrent one-period-ahead forecast is defined in Definition 3.1 below.

Definition 3.1 Recurrent one-period-ahead forecast.
A recurrent one-period-ahead forecast is defined by,

$$\hat{f}_{N+k} = g\left(f_1,\ldots,f_N,\hat{f}_{N+1},\ldots,\hat{f}_{N+k-1}\right), \tag{3.2}$$

and is therefore a function of time series observations $\{f_t\}_{t=1}^{N}$ and previous recurrent one-period-ahead forecasts $\left(\hat{f}_{N+1},\ldots,\hat{f}_{N+k-1}\right)$. Forecasts are therefore recurrent in the sense that two-period-ahead, or further ahead, forecasts are generated using previous forecasts.

Essentially, a recurrent one-period-ahead forecast entails a continuation of the series $\{f_t\}_{t=1}^{N}$ for a single period. When a number of recurrent one-period-ahead forecasts are generated, the situation can be likened to the extrapolation of a signal as a piecewise smooth curve.

Section 3.2.1 below is devoted to the recurrent one-period-ahead forecasting algorithm, which was proposed by Danilov (1997).

3.2.1 Recurrent one-period-ahead forecasting algorithm due to Danilov (1997)

Following the paper by Buchstaber (1994), Danilov (1997) is the first reference to introduce an algorithm for recurrent one-period-ahead forecasting. His proposal is considered in Algorithm 3.1 below. Note that this algorithm has been adapted from noise-contaminated signals to handle a noise-free signal series, which is governed by an LRF.

Algorithm 3.1 Danilov's recurrent one-period-ahead forecasting algorithm.

Suppose that a time series $\{f_t\}_{t=1}^{N}$ was sampled from the broad class of functions (3.1), proposed by Buchstaber (1994). More generally, assume that an LRF governs the observed series.

A one-period-ahead forecast $\left(\hat{f}_{N+1}\right)$ for the series, is a continuation of the series for a single period of time. The forecast is generated using the following steps.

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$.

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}}$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V\Lambda V}'\right)$.

(e) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_r\end{bmatrix}$.

(f) The following system of linear equations is then solved with respect to h_j

$$\begin{aligned} f_{N-\tau+2} &= \frac{1}{N-\tau+2}\sum_{t=1}^{N-\tau+2} f_t + \sum_{j=1}^{r} h_j v_{1,j} \\ &\vdots \\ f_N &= \frac{1}{N-\tau+2}\sum_{t=\tau-1}^{N} f_t + \sum_{j=1}^{r} h_j v_{\tau-1,j} \end{aligned} \tag{3.3}$$

for $j = 1,\ldots,r$, where v_{ij} is the $(i,j)^{th}$ element of matrix $\mathbf{V}$.

(g) The one-period-ahead forecast is calculated using

$$\hat{f}_{N+1} = \frac{1}{N-\tau+1}\sum_{t=\tau}^{N} f_t + \frac{N-\tau+2}{N-\tau+1}\sum_{j=1}^{r} h_j^* v_{\tau,j}\,, \tag{3.4}$$

where a solution of the linear equations in (3.3) is given by $h_1^*,\ldots,h_r^*$.

Proof of (3.4).

The column vectors $\left(\mathbf{x}_j,\ j = 1,\ldots,n\right)$ of the trajectory matrix $\mathbf{X}$ reside on a single r-flat

$$H_r = \left\{\ \mathbf{x} : \mathbf{x} = \overline{\mathbf{x}} + a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \ldots + a_r\mathbf{v}_r;\ \ a_i \in \mathbb{R};\ \ \mathbf{v}_i \in \mathbb{R}^{\tau}; \tau \geq r + 1\right\}, \quad (3.5)$$

where $\overline{\mathbf{x}}$ is the shift vector of the r-flat, and the vectors $\mathbf{v}_1,\ldots,\mathbf{v}_r$ form a base for the parallel linear subspace $\mathfrak{L}_r$.

The column vectors $\left(\tilde{\mathbf{x}}_j,\ j = 1,\ldots,n\right)$ of the centred trajectory matrix $\tilde{\mathbf{X}}$ reside on the parallel linear subspace $\mathfrak{L}_r$. The j^{th} column vector of the centred trajectory matrix can therefore be written as a linear combination of the spanning vectors, i.e.

$$\tilde{\mathbf{x}}_j = \sum_{i=1}^{r} t_i^{(j)}\mathbf{v}_i\,, \quad (3.6)$$

where $\left\{t_1^{(j)},\ldots,t_r^{(j)} \middle| j = 1,\ldots,n\right\}$ are constants. The spanning vectors in (3.6) are eigenvectors, which are obtained through SVD of the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

The centred trajectory matrix has the form,

$$
\begin{aligned}
\underset{(\tau\times n)}{\tilde{\mathbf{X}}} &= \mathbf{X} - \bar{\mathbf{x}}\mathbf{1}' \\
&= \begin{bmatrix} f_1 - \bar{x}_1 & f_2 - \bar{x}_1 & \cdots & f_n - \bar{x}_1 \\ f_2 - \bar{x}_2 & f_3 - \bar{x}_2 & \cdots & f_{n+1} - \bar{x}_2 \\ \vdots & \vdots & \vdots & \vdots \\ f_\tau - \bar{x}_\tau & f_{\tau+1} - \bar{x}_\tau & \cdots & f_N - \bar{x}_\tau \end{bmatrix} \\
&= \begin{bmatrix} \tilde{f}_1 & \tilde{f}_2 & \cdots & \tilde{f}_n \\ \tilde{f}_2 & \tilde{f}_3 & \cdots & \tilde{f}_{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{f}_\tau & \tilde{f}_{\tau+1} & \cdots & \tilde{f}_N \end{bmatrix} \\
&= \begin{bmatrix} \tilde{\mathbf{x}}_1 & \tilde{\mathbf{x}}_2 & \cdots & \tilde{\mathbf{x}}_n \end{bmatrix}.
\end{aligned}
\tag{3.7}
$$

To obtain a one-period-ahead forecast, the centred trajectory matrix in (3.7) is expanded by one column vector, cf. (3.8) below. This actually implies an expansion of the trajectory matrix with a single column vector, to provide for observation f_{N+1}, which will turn out to be the one-period-ahead forecast. The centred trajectory matrix, expanded by a single column vector, is given by

$$
\begin{aligned}
\underset{\tau\times(n+1)}{\tilde{\mathbf{X}}^+} = \begin{bmatrix} \tilde{f}_1 & \tilde{f}_2 & \cdots & \tilde{f}_n & \tilde{f}_{n+1} \\ \tilde{f}_2 & \tilde{f}_3 & \cdots & \tilde{f}_{n+1} & \tilde{f}_{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{f}_\tau & \tilde{f}_{\tau+1} & \cdots & \tilde{f}_N & \tilde{\hat{f}}_{N+1} \end{bmatrix} \\
= \begin{bmatrix} \tilde{\mathbf{x}}_1 & \tilde{\mathbf{x}}_2 & \cdots & \tilde{\mathbf{x}}_n & \tilde{\mathbf{x}}_{n+1} \end{bmatrix},
\end{aligned}
\tag{3.8}
$$

where the one-period-ahead forecast appears in centred form as $\tilde{\hat{f}}_{N+1} = \hat{f}_{N+1} - \bar{x}_\tau^+$. In this representation $\bar{x}_\tau^+$ represents the τ^{th} element belonging to the vector of means $\bar{\mathbf{x}}^+ = \frac{1}{n+1}\mathbf{X}^+\mathbf{1}_{n+1}$, which takes the additional column vector into consideration. This implies that the expanded trajectory matrix, due to inclusion of the one-period-ahead forecast in its column vectors, is denoted by $\mathbf{X}^+ = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n & \mathbf{x}_{n+1} \end{bmatrix}$. Note that the j^{th} element of the vector of means, viewed in terms of the original time series observations, is given by

$$\begin{aligned} \overline{x}^{+} &= \frac{1}{n+1}\sum_{t=j}^{n+j} f_t \\ &= \frac{1}{N-\tau+2}\sum_{t=j}^{N-\tau+j+1} f_t \ . \end{aligned} \tag{3.9}$$

The "centred one-period-ahead " forecast is obtained under the assumption that the column vector, $\tilde{\mathbf{x}}_{n+1}$ in (3.8), also reside on the parallel linear subspace $\mathfrak{L}_r$. This implies that the mentioned column vector can also be written as a linear combination of the spanning vectors of parallel linear subspace $\mathfrak{L}_r$, i.e. that

$$\tilde{\mathbf{x}}_{n+1} = \sum_{j=1}^{r} t_j^{(n+1)}\mathbf{v}_j, \tag{3.10}$$

where $t_1^{(n+1)},\dots,t_r^{(n+1)}$ are unknowns which must be solved. The first $(N-n)$ equations in (3.10) are given by,

$$\begin{aligned} \tilde{f}_{N-\tau+2} &= t_1^{(n+1)}v_{1,1} + t_2^{(n+1)}v_{1,2} + \dots + t_r^{(n+1)}v_{1,r} \\ \tilde{f}_{N-\tau+3} &= t_1^{(n+1)}v_{2,1} + t_2^{(n+1)}v_{2,2} + \dots + t_r^{(n+1)}v_{2,r} \\ &\vdots \\ \tilde{f}_N &= t_1^{(n+1)}v_{\tau-1,1} + t_2^{(n+1)}v_{\tau-1,2} + \dots + t_r^{(n+1)}v_{\tau-1,r} \ . \end{aligned} \tag{3.11}$$

The above system of equations is, however, still in terms of the centred time series observations, i.e. $\tilde{f}_{N-\tau+2},\dots,\tilde{f}_N$. To rectify the situation, the appropriate elements of the vector of means are added, yielding the following system of equations,

$$\begin{aligned}
f_{N-\tau+2} &= \overline{x}_1 + t_1^{(n+1)}v_{1,1} + t_2^{(n+1)}v_{1,2} + \ldots + t_r^{(n+1)}v_{1,r} \\
f_{N-\tau+3} &= \overline{x}_2 + t_1^{(n+1)}v_{2,1} + t_2^{(n+1)}v_{2,2} + \ldots + t_r^{(n+1)}v_{2,r} \\
&\vdots \\
f_N &= \overline{x}_{\tau-1} + t_1^{(n+1)}v_{\tau-1,1} + t_2^{(n+1)}v_{\tau-1,2} + \ldots + t_r^{(n+1)}v_{\tau-1,r}
\end{aligned} \tag{3.12}$$

or alternatively,

$$\begin{aligned}
f_{N-\tau+2} &= \frac{1}{N-\tau+2}\sum_{t=1}^{N-\tau+2} f_t + t_1^{(n+1)}v_{1,1} + t_2^{(n+1)}v_{1,2} + \ldots + t_r^{(n+1)}v_{1,r} \\
f_{N-\tau+3} &= \frac{1}{N-\tau+2}\sum_{t=2}^{N-\tau+3} f_t + t_1^{(n+1)}v_{2,1} + t_2^{(n+1)}v_{2,2} + \ldots + t_r^{(n+1)}v_{2,r} \\
&\vdots \\
f_N &= \frac{1}{N-\tau+2}\sum_{t=\tau-1}^{N} f_t + t_1^{(n+1)}v_{\tau-1,1} + t_2^{(n+1)}v_{\tau-1,2} + \ldots + t_r^{(n+1)}v_{\tau-1,r}.
\end{aligned} \tag{3.13}$$

The system of equations in (3.13) involves $(\tau - 1)$ equations in r unknowns, i.e. the values $t_1^{(n+1)},\ldots,t_r^{(n+1)}$. To adapt a more convenient notation, let

$$h_i = t_i^{(n+1)}, \; i = 1,\ldots,r. \tag{3.14}$$

Using the notation in (3.14), the system of equations in (3.13) becomes,

$$\begin{aligned}
f_{N-\tau+2} &= \frac{1}{N-\tau+2}\sum_{t=1}^{N-\tau+2} f_t + \sum_{j=1}^{r} h_j v_{1,j} \\
f_{N-\tau+3} &= \frac{1}{N-\tau+2}\sum_{t=2}^{N-\tau+3} f_t + \sum_{j=1}^{r} h_j v_{2,j} \\
&\vdots \\
f_N &= \frac{1}{N-\tau+2}\sum_{t=\tau-1}^{N} f_t + \sum_{j=1}^{r} h_j v_{\tau-1,j}
\end{aligned} \tag{3.15}$$

which must be solved for $h_1,\ldots,h_r$. Suppose that $h_1^*,\ldots,h_r^*$ is a solution. The one-period-ahead forecast can then be written as,

$$f_{N+1} = \frac{1}{N-\tau+2}\sum_{t=\tau}^{N+1} f_t + h_1^* v_{\tau,1} + h_2^* v_{\tau,2} + \ldots + h_r^* v_{\tau,r} \, . \tag{3.16}$$

Solving (3.16) for f_{N+1} gives a one-period-ahead forecast; denote this by $\hat{f}_{N+1}$. This one-period-ahead forecast is then given by,

$$\hat{f}_{N+1} = \frac{1}{(N-\tau+1)}\sum_{t=\tau}^{N} f_i + \frac{(N-\tau+2)}{(N-\tau+1)}\sum_{i=1}^{r} h_j^* v_{\tau,j} \quad . \tag{3.17}$$

This concludes the proof. □

Remark 3.1

(a) The proof of Danilov's recurrent one-period-ahead forecasting method is provided above, to highlight the difference with the approach that we will define in sections that follow.

(b) It is clear that Algorithm 3.1 above assumes that both the window length (τ) and the number of eigenvectors to use (r) are known. It must be re-iterated that the algorithm above, was adapted to handle the extrapolation of noise-free signals. When noise-free signals are considered, all column vectors of a trajectory matrix reside on a single r-flat. SSA forecasting, in such a scenario, entails a non-parametric extrapolation of the signal. In the case of noise-contaminated signals, the situation is not as simple, and many other considerations need to be taken into account. These issues will be elaborated upon in Section 3.3 below.

3.2.2 Recurrent one-period-ahead forecasting algorithm due to Venter (1998)

Venter (1998) introduced a more elegant formulation of a recurrent one-period-ahead forecast, using a coordinate free approach. The approach generates recurrent one-period-ahead forecasts, by applying a projection onto a parallel linear subspace and using a shift back to the original r-flat. The approach is much more direct, and therefore does not require the solving of a system of equations, as is the case in Danilov's approach. The algorithm due to Venter (1998) is therefore well suited for packages such as MATLAB, which prefer a vectorised approach to problem solving.

The approach due to Venter (1998) is supplied in the form of Algorithm 3.2 below.

Algorithm 3.2 Venter's recurrent one-period-ahead forecasting algorithm.

Suppose that a time series $\{f_t\}_{t=1}^{N}$ was sampled from the broad class of functions in (3.1) and that the series is therefore governed by an LRF of rank r.

The one-period-ahead forecast $\left(\hat{f}_{N+1}\right)$ for the series, is also considered a continuation of the series for a single period of time, in this algorithm. The forecast is generated using the following steps.

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $r < \tau \leq n (= N - \tau + 1)$.

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}}$, using the shift vector $\bar{\mathbf{x}} = \frac{1}{n}\mathbf{X1}$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}'\right)$.

(e) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_r\end{bmatrix}$.

(f) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r} = \mathbf{VV}'$ of the parallel linear subspace $\mathfrak{L}_r$.

(g) Construct the one-period-ahead forecast, using

$$\hat{f}_{N+1} = \bar{x}_\tau + \frac{\left\langle \mathbf{P}_{\mathfrak{L}_r} \tilde{\mathbf{x}}_{n+1}^{(0,\tau)}, \mathbf{e}_\tau \right\rangle}{1 - \left\| \mathbf{P}_{\mathfrak{L}_r} \mathbf{e}_\tau \right\|^2} \tag{3.18}$$

where

$$\begin{aligned} \tilde{\mathbf{x}}_{n+1}^{(0,\tau)} &= (n+1)^{\text{th}} \text{ column vector of matrix } \tilde{\mathbf{X}}^+ \text{ with zero in position } \tau \\ \mathbf{e}_\tau &= \tau^{th} \text{ column vector of the } \tau \times \tau \text{ identity matrix } \mathbf{I}_\tau \ . \end{aligned} \tag{3.19}$$

Proof.

The column vectors of the trajectory matrix $(\mathbf{X})$ resides on a single r-flat of the form,

$$H = \left\{ \mathbf{x} : \mathbf{x} = \underbrace{\bar{\mathbf{x}}}_{\textbf{shift vector}} + \underbrace{a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \ldots + a_r\mathbf{v}_r}_{\in \mathfrak{L}_r} ;\ a_i \in \mathbb{R};\ \mathbf{v}_i \in \mathbb{R}^\tau ; \tau \geq r+1 \right\}. \tag{3.20}$$

The column vectors of the trajectory matrix are now expanded to include $\hat{f}_{N+1}$, which is the one-period-ahead forecast. The expanded trajectory matrix now has the form,

$$\begin{aligned} \underset{\tau\times(n+1)}{\mathbf{X}^+} &= \begin{bmatrix} f_1 & f_2 & \cdots & f_n & f_{n+1} \\ f_2 & f_3 & \cdots & f_{n+1} & f_{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_\tau & f_{\tau+1} & \cdots & f_N & \hat{f}_{N+1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n & \mathbf{x}_{n+1} \end{bmatrix}. \end{aligned} \tag{3.21}$$

The one-period-ahead forecast is constructed under the assumption that the expanded column vector, $\mathbf{x}_{n+1}$, also resides on the r-flat in (3.20). This implies that the expanded centred time delay matrix column vector, $\tilde{\mathbf{x}}_{n+1}$, resides on the parallel linear subspace $\mathfrak{L}_r$.

The column vector $\tilde{\mathbf{x}}_{n+1}$ is now partitioned as follows,

$$\begin{aligned}\underset{(\tau\times1)}{\tilde{\mathbf{x}}_{n+1}} &= \begin{bmatrix}\tilde{f}_{n+1}\\ \vdots\\ \tilde{f}_N\\ 0\end{bmatrix} + \begin{bmatrix}0\\ \vdots\\ 0\\ \hat{\tilde{f}}_{N+1}\end{bmatrix}\\ &= \tilde{\mathbf{x}}_{n+1}^{(0,\tau)} + \hat{\tilde{f}}_{N+1}\mathbf{e}_\tau\end{aligned} \tag{3.22}$$

where the one-period-ahead forecast appears in centred form as $\hat{\tilde{f}}_{N+1} = \hat{f}_{N+1} - \overline{x}_\tau$, and $\mathbf{e}_\tau$ is the τ^{th} column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$.

Let $\mathbf{P}_{\mathfrak{L}_r^\perp}$ now denote the projection matrix of the linear subspace orthogonal to the parallel linear subspace $\mathfrak{L}_r$, i.e. $\mathbf{P}_{\mathfrak{L}_r^\perp} = \mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}$.

It then follows that

$$\mathbf{P}_{\mathfrak{L}_r^\perp}\tilde{\mathbf{x}}_{n+1} = \mathbf{0}, \tag{3.23}$$

since $\tilde{\mathbf{x}}_{n+1} \in \mathfrak{L}_r$ and $\mathfrak{L}_r^\perp \perp \mathfrak{L}_r$. Combining the partitioning in (3.22) with (3.23), yields

$$\begin{aligned}\mathbf{P}_{\mathfrak{L}_r^\perp}\left(\tilde{\mathbf{x}}_{n+1}^{(0,\tau)} + \hat{\tilde{f}}_{N+1}\mathbf{e}_\tau\right) &= \mathbf{0}\\ \hat{\tilde{f}}_{N+1}\mathbf{P}_{\mathfrak{L}_r^\perp}\mathbf{e}_\tau &= -\mathbf{P}_{\mathfrak{L}_r^\perp}\tilde{\mathbf{x}}_{n+1}^{(0,\tau)}.\end{aligned} \tag{3.24}$$

Multiplying (3.24) throughout with the vector $\left(\mathbf{P}_{\mathfrak{L}_r^\perp}\mathbf{e}_\tau\right)'$, yields

$$\begin{aligned}\hat{\tilde{f}}_{N+1}\left(\mathbf{P}_{\mathfrak{L}_r^\perp}\mathbf{e}_\tau\right)'\left(\mathbf{P}_{\mathfrak{L}_r^\perp}\mathbf{e}_\tau\right) &= -\left(\mathbf{P}_{\mathfrak{L}_r^\perp}\mathbf{e}_\tau\right)'\left(\mathbf{P}_{\mathfrak{L}_r^\perp}\tilde{\mathbf{x}}_{n+1}^{(0,\tau)}\right)\\ \hat{\tilde{f}}_{N+1}\left\|\mathbf{P}_{\mathfrak{L}_r^\perp}\mathbf{e}_\tau\right\|^2 &= -\left(\mathbf{P}_{\mathfrak{L}_r^\perp}\mathbf{e}_\tau\right)'\left(\mathbf{P}_{\mathfrak{L}_r^\perp}\tilde{\mathbf{x}}_{n+1}^{(0,\tau)}\right).\end{aligned} \tag{3.25}$$

It is clear that the squared norm $\left\| \mathbf{P}_{\mathfrak{L}_r^{\perp}} \mathbf{e}_\tau \right\|^2$ in (3.25) is a scalar and division of (3.25) throughout therefore yields,

$$\begin{aligned} \hat{\tilde{f}}_{N+1} &= -\frac{\left(\mathbf{P}_{\mathfrak{L}_r^{\perp}} \mathbf{e}_\tau\right)' \left(\mathbf{P}_{\mathfrak{L}_r^{\perp}} \tilde{\mathbf{x}}_{n+1}^{(0,\tau)}\right)}{\left\| \mathbf{P}_{\mathfrak{L}_r^{\perp}} \mathbf{e}_\tau \right\|^2} \\ \hat{\tilde{f}}_{N+1} &= -\frac{\mathbf{e}_\tau' \mathbf{P}_{\mathfrak{L}_r^{\perp}}' \mathbf{P}_{\mathfrak{L}_r^{\perp}} \tilde{\mathbf{x}}_{n+1}^{(0,\tau)}}{\left\| \mathbf{P}_{\mathfrak{L}_r^{\perp}} \mathbf{e}_\tau \right\|^2}. \end{aligned} \tag{3.26}$$

The projection matrix $\mathbf{P}_{\mathfrak{L}_r^{\perp}}$ is symmetric and idempotent, hence

$$\begin{aligned} \hat{\tilde{f}}_{N+1} &= -\frac{\mathbf{e}_\tau' \mathbf{P}_{\mathfrak{L}_r^{\perp}} \tilde{\mathbf{x}}_{n+1}^{(0,\tau)}}{\left\| \mathbf{P}_{\mathfrak{L}_r^{\perp}} \mathbf{e}_\tau \right\|^2} \\ \hat{\tilde{f}}_{N+1} &= -\frac{\left\langle \mathbf{P}_{\mathfrak{L}_r^{\perp}} \tilde{\mathbf{x}}_{n+1}^{(0,\tau)}, \mathbf{e}_\tau \right\rangle}{\left\| \mathbf{P}_{\mathfrak{L}_r^{\perp}} \mathbf{e}_\tau \right\|^2}. \end{aligned} \tag{3.27}$$

The fact that, $\mathbf{P}_{\mathfrak{L}_r^{\perp}} = \mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}$, is now applied to (3.27). This, with some simplification, yields

$$\begin{aligned}
\hat{\hat{f}}_{N+1} &= -\frac{\left\langle \left(\mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}\right)\tilde{\mathbf{x}}_{n+1}^{(0,\tau)}, \mathbf{e}_\tau \right\rangle}{\left\|\left(\mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}\right)\mathbf{e}_\tau\right\|^2} \\
&= -\frac{\left\langle \left(\mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}\right)\mathbf{e}_\tau, \tilde{\mathbf{x}}_{n+1}^{(0,\tau)} \right\rangle}{\left\|\left(\mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}\right)\mathbf{e}_\tau\right\|^2} \\
&= -\frac{\left\langle \mathbf{e}_\tau - \mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau, \tilde{\mathbf{x}}_{n+1}^{(0,\tau)} \right\rangle}{\left\|\left(\mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}\right)\mathbf{e}_\tau\right\|^2} \\
&= \frac{\left\langle \mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau, \tilde{\mathbf{x}}_{n+1}^{(0,\tau)} \right\rangle}{\left\|\left(\mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}\right)\mathbf{e}_\tau\right\|^2} \\
&= \frac{\left\langle \mathbf{P}_{\mathfrak{L}_r}\tilde{\mathbf{x}}_{n+1}^{(0,\tau)}, \mathbf{e}_\tau \right\rangle}{\left\|\left(\mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}\right)\mathbf{e}_\tau\right\|^2}.
\end{aligned} \tag{3.28}$$

Finally, it follows that

$$\begin{aligned}
\hat{\hat{f}}_{N+1} &= \frac{\left\langle \mathbf{P}_{\mathfrak{L}_r}\tilde{\mathbf{x}}_{n+1}^{(0,\tau)}, \mathbf{e}_\tau \right\rangle}{\left\|\left(\mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}\right)\mathbf{e}_\tau\right\|^2} \\
&= \frac{\left\langle \mathbf{P}_{\mathfrak{L}_r}\tilde{\mathbf{x}}_{n+1}^{(0,\tau)}, \mathbf{e}_\tau \right\rangle}{\left\|\mathbf{e}_\tau - \mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau\right\|^2} \\
&= \frac{\left\langle \mathbf{P}_{\mathfrak{L}_r}\tilde{\mathbf{x}}_{n+1}^{(0,\tau)}, \mathbf{e}_\tau \right\rangle}{1 - \left\|\mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau\right\|^2}.
\end{aligned} \tag{3.29}$$

This concludes the proof. □

Remark 3.2

(a) The proof given above is essentially that of Venter (1998).

(b) It is clear from (3.18) that erroneous forecasting will result when $\left\|\mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau\right\| \approx 1$. This situation gives rise to a condition termed the *verticality condition*, cf. Golyandina *et al.* (2001, p.95). A recurrent one-period-ahead forecast does not exist when $\mathbf{e}_\tau \in H_r$ or, alternatively, when $\mathbf{e}_\tau \in \mathfrak{L}_r$. This implies that the r-flat, spanned by the column vectors of the trajectory matrix, may not be a vertical space. It is therefore imperative that this condition be guarded against when producing forecasts. Attention is given to this issue in Section 3.7 below.

3.2.3 An alternative one-period-ahead forecasting algorithm

The algorithm that follows builds on the ideas outlined by Venter (1998) and is an alternative recurrent one-period-ahead forecasting algorithm. The formulation that is proposed can easily be adapted to handle joint-horizon k-period-ahead forecasting. The formulation also uses a coordinate-free approach, and is therefore in vectorised form.

Algorithm 3.3 An alternative recurrent one-period-ahead forecasting algorithm.

Suppose that a time series $\{f_t\}_{t=1}^{N}$ was sampled from the broad class of functions in (3.1) and that it is governed by an LRF of rank r.

A recurrent one-period-ahead forecast is constructed using the following steps.

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $r < \tau \leq n (= N - \tau + 1)$.

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}}$, using the shift vector $\overline{\mathbf{x}} = \frac{1}{n}\mathbf{X}\mathbf{1}$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'\right)$.

(e) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$.

(f) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r} = \mathbf{V}\mathbf{V}'$ of the linear subspace $\mathfrak{L}_r$.

(g) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r^{\perp}} = \mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}$ of the orthogonal linear subspace $\mathfrak{L}_r^{\perp}$.

(h) Partition the projection matrix in (g) as follows

$$\begin{aligned} \mathbf{P}_{\mathfrak{L}_r^{\perp}} &= \left[\mathbf{P}_{\mathfrak{L}_r^{\perp}} \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix} \middle| \mathbf{P}_{\mathfrak{L}_r^{\perp}} \mathbf{e}_\tau \right] \\ &= \left[\underset{\tau\times(\tau-1)}{\mathbf{P}_1} \middle| \underset{\tau\times 1}{\mathbf{P}_2} \right], \end{aligned} \tag{3.30}$$

where $\mathbf{e}_j$ is the j^{th} column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$.

(i) Calculate the one-period-ahead forecast using

$$\hat{f}_{N+1} = \overline{x}_\tau - \left(\mathbf{P}_2'\mathbf{P}_2\right)^{-1}\left(\mathbf{P}_2'\mathbf{P}_1\right)\begin{bmatrix} f_{N-\tau+2} - \overline{x}_1 \\ \vdots \\ f_N - \overline{x}_{\tau-1} \end{bmatrix}, \tag{3.31}$$

where

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{P}_{\mathfrak{L}_r^\perp}\begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix} \\ \mathbf{P}_2 &= \mathbf{P}_{\mathfrak{L}_r^\perp}\mathbf{e}_\tau \ . \end{aligned} \tag{3.32}$$

Proof.

The column vectors of the trajectory matrix $(\mathbf{X})$ reside on a single r-flat of the form,

$$H = \left\{ \mathbf{x} : \mathbf{x} = \underbrace{\overline{\mathbf{x}}}_{\textbf{shift vector}} + \underbrace{a_1\mathbf{v}_1 + \ldots + a_r\mathbf{v}_r}_{\in \mathfrak{L}_r};\ a_i \in \mathbb{R};\ \mathbf{v}_i \in \mathbb{R}^\tau; \tau \geq r+1 \right\}. \tag{3.33}$$

The column vectors of the trajectory matrix are now expanded to include $\hat{f}_{N+1}$, which is the one-period-ahead forecast. The expanded trajectory matrix now has the form

$$\begin{aligned} \underset{\tau\times(n+1)}{\mathbf{X}^+} &= \begin{bmatrix} f_1 & f_2 & \cdots & f_n & f_{n+1} \\ f_2 & f_3 & \cdots & f_{n+1} & f_{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_\tau & f_{\tau+1} & \cdots & f_N & \hat{f}_{N+1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n & \mathbf{x}_{n+1} \end{bmatrix}. \end{aligned} \tag{3.34}$$

The one-period-ahead forecast is constructed under the assumption that the expanded column vector, $\mathbf{x}_{n+1}$, also resides on the r-flat in (3.33), i.e. that $\mathbf{x}_{n+1} \in H_r$. This implies that the expanded centred time delay matrix column vector, $\tilde{\mathbf{x}}_{n+1}$, resides on the parallel linear subspace $\mathfrak{L}_r$, i.e. that $\tilde{\mathbf{x}}_{n+1} \in \mathfrak{L}_r$.

Let $\mathbf{P}_{\mathfrak{L}_r^{\perp}}$ now denote the projection matrix of the linear subspace orthogonal to the parallel linear subspace $\mathfrak{L}_r$, i.e. $\mathbf{P}_{\mathfrak{L}_r^{\perp}} = \mathbf{I}_{\tau} - \mathbf{P}_{\mathfrak{L}_r}$.

It then follows that

$$\mathbf{P}_{\mathfrak{L}_r^{\perp}} \tilde{\mathbf{x}}_{n+1} = \mathbf{0}. \qquad (3.35)$$

From this and (3.30) it follows that

$$[\mathbf{P}_1 | \mathbf{P}_2] \tilde{\mathbf{x}}_{n+1} = \mathbf{0}. \qquad (3.36)$$

It is clear that (3.36) can be written as

$$\left[\mathbf{P}_1 \mid \mathbf{P}_2\right] \begin{bmatrix} f_{N-\tau+2} - \overline{x}_1 \\ \vdots \\ f_N - \overline{x}_{\tau-1} \\ \hat{f}_{N+1} - \overline{x}_{\tau} \end{bmatrix} = \mathbf{0}. \qquad (3.37)$$

In (3.37) above, $\hat{f}_{N+1}$ denotes the one-period-ahead forecast, which must be determined.

The following few steps are then followed

$$
\begin{aligned}
\left(\hat{f}_{N+1} - \overline{x}_{\tau}\right)\mathbf{P}_2 &= -\mathbf{P}_1\begin{bmatrix} f_{N-\tau+2} - \overline{x}_1 \\ \vdots \\ f_N - \overline{x}_{\tau-1} \end{bmatrix} \\
\left(\hat{f}_{N+1} - \overline{x}_{\tau}\right) &= -\left(\mathbf{P}_2'\mathbf{P}_2\right)^{-1}\left(\mathbf{P}_2'\mathbf{P}_1\right)\begin{bmatrix} f_{N-\tau+2} - \overline{x}_1 \\ \vdots \\ f_N - \overline{x}_{\tau-1} \end{bmatrix} \qquad (3.38) \\
\hat{f}_{N+1} &= \overline{x}_{\tau} - \left(\mathbf{P}_2'\mathbf{P}_2\right)^{-1}\left(\mathbf{P}_2'\mathbf{P}_1\right)\begin{bmatrix} f_{N-\tau+2} - \overline{x}_1 \\ \vdots \\ f_N - \overline{x}_{\tau-1} \end{bmatrix}
\end{aligned}
$$

which yields the one-period-ahead forecast, and the proof is therefore concluded. □

It is clear from the above proof that (3.37) is just a system of equations in vectorised form. The system of equations is indirectly solved using a coordinate-free approach, i.e. via projections. The approach therefore indirectly uses elements of the approaches proposed by Danilov (1997) and Venter (1998). Algorithm 3.3 partitions the projection matrix $\mathbf{P}_{\mathfrak{L}_r^{\perp}}$, where Algorithm 3.2 partitions the vector $\tilde{\mathbf{x}}_{n+1}$. This causes the formulation in Algorithm 3.3 to be more general in the sense that it can be extended to handle joint-horizon k-period-ahead forecasts and can also be directly applied in the multi-channel SSA case.

Remark 3.3

It is clear from (3.31) that a recurrent one-period-ahead forecast exists only if $\left(\mathbf{P}_2'\mathbf{P}_2\right)$ is of full rank. This is equivalent to stating that $\mathbf{e}_{\tau} \notin \mathfrak{L}_r$ where $\mathfrak{L}_r = span\left(\mathbf{v}_1 \quad \cdots \quad \mathbf{v}_r\right)$. The verticality condition must therefore also hold, for the recurrent one-period-ahead forecast to exist.

3.2.4 Recurrent one-period-ahead forecasting algorithm due to Golyandina *et al.* (2001)

Golyandina *et al.* (2001) also proposed a recurrent one-period-ahead forecasting algorithm which they coined Basic SSA R-forecasting. Their approach builds on the notion of LRFs.

Their approach appears in the form of Algorithm 3.4 below. The algorithm has been adapted for the noise-free case, since their original algorithm provides for a noise-contaminated series.

Two algorithms are entertained below. Algorithm 3.4 does not make use of centring. The trajectory matrix is therefore not centred, by subtracting the vector of means from each column vector.

Algorithm 3.4 Basic SSA R-forecasting due to Golyandina et al. (2001) when no centring is used.

Suppose that a time series $\{f_t\}_{t=1}^{N}$, which is governed by an LRF of rank r, is observed.

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $r < \tau \leq n(= N - \tau + 1)$.

(b) The trajectory matrix is not centred and the resulting scatter matrix is $\mathbf{XX}'$.

(c) Perform SVD of the scatter matrix $\left(\mathbf{XX}' = \mathbf{V\Lambda V}'\right)$.

(d) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$.

(e) Form the vectors $\left\{ \mathbf{v}_j^{\nabla} = \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}' \mathbf{v}_j \middle| j = 1,\ldots,r \right\}$.

(f) Form the scalar $\nu^2 = \sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right)^2$ where $\mathbf{e}_\tau$ is the last column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and $\mathbf{v}_j$ is the j^{th} eigenvector derived in step (c).

(g) Calculate the one-period-ahead forecast using

$$\hat{f}_{N+1} = a_1 f_N + a_2 f_{N-1} + \ldots + a_{\tau-1} f_{N-\tau+2}, \tag{3.39}$$

where

$$\begin{bmatrix} a_{\tau-1} \\ \vdots \\ a_1 \end{bmatrix} = \frac{1}{1-\nu^2}\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\mathbf{v}_j^\nabla . \qquad (3.40)$$

Proof.

See Golyandina *et al.* (2001, pp. 243-248).

Remark 3.4

Golyandina *et al.* (2001, p.95) state that a recurrent one-period-ahead forecast exists only if the verticality condition is met, i.e. that $\mathbf{e}_\tau \notin span\left(\mathbf{v}_1 \quad \cdots \quad \mathbf{v}_r\right)$.

It is clear from (3.39) that the one-period-ahead forecast is written in the form of an LRF. There are a few additional benefits that arise when using this representation. The coefficients in the LRF are of importance, since they must reside within a unit circle. It is therefore possible to assess the quality of forecasts by assessing stability in terms of the coefficients $a_1,\ldots,a_r$. Golyandina *et al.* (2001) can be consulted in this regard.

Algorithm 3.5 below uses centring of the trajectory matrix. This result is equivalent to the algorithm due to Venter (1998) and a proof is provided in Section 3.2.5.

Algorithm 3.5 Basic SSA R-forecasting due to Golyandina et al. (2001) when centring is used.

Suppose that an observed time series $\{f_t\}_{t=1}^{N}$ is governed by an LRF of rank r and therefore from the broad class of functions in (3.1).

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $r < \tau \leq n (= N - \tau + 1)$.

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}}$, using the shift vector $\overline{\mathbf{x}} = \frac{1}{n}\mathbf{X1}$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V\Lambda V}'\right)$.

(e) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$.

(f) Form the vectors $\left\{ \mathbf{v}_j^{\nabla} = \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}' \mathbf{v}_j \middle| j = 1,\ldots,r \right\}$.

(g) Form the scalar $\nu^2 = \sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right)^2$ where $\mathbf{e}_\tau$ is the last column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and $\mathbf{v}_j$ is the j^{th} eigenvector derived in step (c).

(h) Calculate the one-period-ahead forecast using

$$\hat{f}_{N+1} = \bar{x}_\tau + a_1 \left(f_N - \bar{x}_{\tau-1} \right) + a_2 \left(f_{N-1} - \bar{x}_{\tau-2} \right) + \ldots + a_{\tau-1} \left(f_{N-\tau+2} - \bar{x}_1 \right), \quad (3.41)$$

where

$$\begin{bmatrix} a_{\tau-1} \\ \vdots \\ a_1 \end{bmatrix} = \frac{1}{1-\nu^2} \sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right) \mathbf{v}_j^{\nabla}. \quad (3.42)$$

3.2.5 Equivalence of the algorithms of Venter (1998) and Golyandina *et al.* (2001).

A proof of the equivalence of Algorithm 3.2 and Algorithm 3.5 above is given in this section. From the proof it becomes evident that the coordinate-free approach and LRF approach produce equivalent results. The algorithms in question use centring. A proof for the case where no centring is used will be similar.

Proof.

Suppose that an observed time series $\{ f_t \}_{t=1}^{N}$ is governed by an LRF and therefore from the broad class in (3.1). The assumption is therefore made that the column vectors of the trajectory matrix, formed using the series, reside on a single r-flat. The trajectory matrix $(\mathbf{X})$ and the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$ are formed. SVD of the scatter matrix is performed, which results in matrix

$$\underset{(\tau \times r)}{\mathbf{V}} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}. \quad (3.43)$$

This matrix has, as column vectors, the orthonormal base vectors of the parallel linear subspace $\mathfrak{L}_r$.

According to Golyandina *et al.* (2001) the one-period-ahead forecast is then given by

$$\hat{f}_{N+1} = \overline{x}_\tau + a_1\left(f_N - \overline{x}_{\tau-1}\right) + a_2\left(f_{N-1} - \overline{x}_{\tau-2}\right) + \ldots + a_{\tau-1}\left(f_{N-\tau+2} - \overline{x}_1\right), \quad (3.44)$$

where

$$\begin{bmatrix} a_{\tau-1} \\ \vdots \\ a_1 \end{bmatrix} = \frac{1}{1-\nu^2}\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\mathbf{v}_j^\nabla . \quad (3.45)$$

In vectorised form, the forecast in (3.44) is given by

$$\hat{f}_{N+1} = \overline{x}_\tau + \begin{bmatrix} a_{\tau-1} & \cdots & a_1 \end{bmatrix}\begin{bmatrix} \left(f_{N-\tau+2} - \overline{x}_1\right) & \cdots & \left(f_N - \overline{x}_{\tau-1}\right) \end{bmatrix}' . \quad (3.46)$$

Substituting (3.45) in (3.46), yields

$$\hat{f}_{N+1} = \overline{x}_\tau + \left[\frac{1}{1-\nu^2}\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\mathbf{v}_j^\nabla\right]'\begin{bmatrix} \left(f_{N-\tau+2} - \overline{x}_1\right) & \cdots & \left(f_N - \overline{x}_{\tau-1}\right) \end{bmatrix}', \quad (3.47)$$

where $\mathbf{v}_j^\nabla$ is a vector containing the first $(\tau-1)$ components of the eigenvector (orthonormal base vector) $\mathbf{v}_j$, i.e.

$$\mathbf{v}_j^\nabla = \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}' \mathbf{v}_j \quad (3.48)$$

for $j = 1,\ldots,r$. Furthermore, the scalar ν^2 in (3.47) is given by

$$\nu^2 = \sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right)^2 . \tag{3.49}$$

Substituting (3.48) and (3.49) into (3.47), yields

$$\hat{f}_{N+1} = \overline{x}_\tau + \frac{\left[\sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right) \left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1} \right]' \mathbf{v}_j \right]' \left[\left(f_{N-\tau+2} - \overline{x}_1 \right) \quad \cdots \quad \left(f_N - \overline{x}_{\tau-1} \right) \right]'}{1 - \sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right)^2} . \tag{3.50}$$

It is clear that the denominator in (3.50) is simply

$$\begin{aligned} 1 - \left\{ \mathbf{e}_\tau' \mathbf{v}_1 \mathbf{v}_1' \mathbf{e}_\tau + \ldots + \mathbf{e}_\tau' \mathbf{v}_r \mathbf{v}_r' \mathbf{e}_\tau \right\} &= 1 - \mathbf{e}_\tau' \left[\mathbf{v}_1 \mathbf{v}_1' \quad \cdots \quad \mathbf{v}_r \mathbf{v}_r' \right] \mathbf{e}_\tau \\ &= 1 - \mathbf{e}_\tau' \mathbf{V} \mathbf{V}' \mathbf{e}_\tau , \end{aligned} \tag{3.51}$$

where matrix $\mathbf{V}$ is defined in (3.43).

If (3.51) is substituted into (3.50), then the one-period-ahead forecast becomes

$$\hat{f}_{N+1} = \overline{x}_\tau + \frac{\left[\sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right) \left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1} \right]' \mathbf{v}_j \right]' \left[\left(f_{N-\tau+2} - \overline{x}_1 \right) \quad \cdots \quad \left(f_N - \overline{x}_{\tau-1} \right) \right]'}{1 - \mathbf{e}_\tau' \mathbf{V} \mathbf{V}' \mathbf{e}_\tau} . \tag{3.52}$$

With attention still affixed on the denominator, recall that the projection matrix of the parallel linear subspace $\mathfrak{L}_r$ is given by $\mathbf{P}_{\mathfrak{L}_r} = \mathbf{V}\mathbf{V}'$. If this is introduced in (3.52), then it results in

$$\begin{aligned}\hat{f}_{N+1} &= \bar{x}_\tau + \frac{\left[\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1}\right]'\mathbf{v}_j\right]'\left[\left(f_{N-\tau+2}-\bar{x}_1\right) \quad \cdots \quad \left(f_N-\bar{x}_{\tau-1}\right)\right]'}{1-\mathbf{e}_\tau'\mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau} \\ &= \bar{x}_\tau + \frac{\left[\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1}\right]'\mathbf{v}_j\right]'\left[\left(f_{N-\tau+2}-\bar{x}_1\right) \quad \cdots \quad \left(f_N-\bar{x}_{\tau-1}\right)\right]'}{1-\left\|\mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau\right\|^2}.\end{aligned}$$

$$(3.53)$$

The last step in (3.53) made use of the fact that a projection matrix is idempotent. The denominator in (3.53) now has the required form and attention is shifted to the numerator.

The part $\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1}\right]'\mathbf{v}_j$ can be written as

$$\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1}\right]'\mathbf{v}_j = \sum_{j=1}^{r}\left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1}\right]'\mathbf{v}_j\left(\mathbf{v}_j'\mathbf{e}_\tau\right). \qquad (3.54)$$

The step in (3.54) makes use of the fact that $\left(\mathbf{e}_\tau'\mathbf{v}_j\right)$ is simply a scalar. The next step is also clear

$$\begin{aligned}\sum_{j=1}^{r}\left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1}\right]'\mathbf{v}_j\left(\mathbf{v}_j'\mathbf{e}_\tau\right) &= \left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1}\right]'\left[\sum_{j=1}^{r}\mathbf{v}_j\mathbf{v}_j'\right]\mathbf{e}_\tau \\ &= \left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1}\right]'\mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau.\end{aligned} \qquad (3.55)$$

Combining (3.55) and the result in (3.53), yields

$$\begin{aligned}\hat{f}_{N+1} &= \bar{x}_\tau + \frac{\left[\begin{bmatrix}\mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1}\end{bmatrix}' \mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau\right]' \left[\left(f_{N-\tau+2}-\bar{x}_1\right) \ \cdots \ \left(f_N-\bar{x}_{\tau-1}\right)\right]'}{1-\left\|\mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau\right\|^2} \\ &= \bar{x}_\tau + \frac{\left(\mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau\right)' \left[\mathbf{e}_1 \ \cdots \ \mathbf{e}_{\tau-1}\right]\left[\left(f_{N-\tau+2}-\bar{x}_1\right) \ \cdots \ \left(f_N-\bar{x}_{\tau-1}\right)\right]'}{1-\left\|\mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau\right\|^2}.\end{aligned} \tag{3.56}$$

It is clear in (3.56) that

$$\left[\mathbf{e}_1 \ \cdots \ \mathbf{e}_{\tau-1}\right]\left[\left(f_{N-\tau+2}-\bar{x}_1\right) \ \cdots \ \left(f_N-\bar{x}_{\tau-1}\right)\right]' = \begin{bmatrix} f_{N-\tau+2}-\bar{x}_1 \\ \vdots \\ f_N-\bar{x}_{\tau-1} \\ 0 \end{bmatrix}. \tag{3.57}$$

Combining (3.57) and (3.56), yields

$$\begin{aligned}\hat{f}_{N+1} &= \bar{x}_\tau + \frac{\left(\mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau\right)'}{1-\left\|\mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau\right\|^2}\begin{bmatrix} f_{N-\tau+2}-\bar{x}_1 \\ \vdots \\ f_N-\bar{x}_{\tau-1} \\ 0 \end{bmatrix} \\ &= \bar{x}_\tau + \frac{\left\langle \mathbf{P}_{\mathfrak{L}_r}\tilde{\mathbf{x}}_{n+1}^{(0,\tau)}, \mathbf{e}_\tau \right\rangle}{1-\left\|\mathbf{P}_{\mathfrak{L}_r}\mathbf{e}_\tau\right\|^2},\end{aligned} \tag{3.58}$$

where $\tilde{\mathbf{x}}_{n+1}^{(0,\tau)} = \left[f_{N-\tau+2}-\bar{x}_1 \ \cdots \ f_N-\bar{x}_{\tau-1} \ \ 0\right]'$. This concludes the proof. □

3.3 Recurrent one-period-ahead forecasting algorithms for noise-contaminated signals

Recurrent one-period-ahead forecasting of noise-free series was considered in Section 3.2 above. Forecasting in the noise-free scenario entails a continuation of the series under consideration. The underlying assumption made in this section is that a noise-contaminated signal series is under consideration, with structure

$$x_t = f_t^{(1)} + f_t^{(2)},\ t = 1,\ldots,N\ , \tag{3.59}$$

where $\{f_t^{(1)}\}_{t=1}^N$ is a noise-free signal series governed by an LRF and therefore from the broad class of functions in (3.1). The series $\{f_t^{(2)}\}_{t=1}^N$ is assumed to be a noise series of independent and identically distributed observations, originating from some unknown distribution with zero expectation and constant variance. The column vectors of the trajectory matrix, based on the signal series $\{f_t^{(1)}\}_{t=1}^N$, reside on a single r-flat H_r. The column vectors of the trajectory matrix, based on the noise-contaminated time series $\{x_t\}_{t=1}^N$, do not reside on the r-flat H_r. Instead, these column vectors represent a multivariate scatter around the mentioned r-flat.

When noise is introduced into a signal series, which is governed by an LRF, then recurrent forecasting will contain two sources of forecasting error. The first recurrent one-period-ahead forecast is solely based on the time series, which involves noise. The second recurrent one-period-ahead forecast is based on the noise-contaminated time series, and the previous one-period-ahead forecast. Two sources of error can be identified in such a case. One source is due to noise, and the second due to forecasting error. This process would therefore have a cumulative effect and forecasts will degenerate at a cumulative rate, when using recurrent forecasting.

Forecasting error is inevitable in the presence of noise. This is accepted as fact and cannot be minimized other than using the "best possible model". If no knowledge regarding the underlying signal series in (3.59) is available, a method must be found to identify the "best possible model". This issue will be addressed in Chapter 4. In this section the assumption is made that the best possible model is known. This translates into perfect knowledge with regard to the window length (τ) and number of eigenvectors (r) to use in the SSA scheme.

As mentioned, noise also influences forecasts. Noise causes the column vectors of a trajectory matrix, based on a noise-contaminated signal, not to reside on a single r-flat. It is therefore a logical step to investigate methods that smooth out noise or somehow reduces the influence thereof. This translates into reconstructing a signal series with certain desired properties. The column vectors of the reconstructed signal series must reside on a single r-flat.

Two approaches will be considered. The first approach uses an approximate series, which is due to Golyandina *et al.* (2001). The second approach, which is a proposal of this study, involves use of a reconstructed signal series, obtained using the Cadzow (1988) iterative property mapping algorithm.

3.3.1 Algorithms using an approximate series

The Basic SSA R-forecasting algorithm, proposed by Golyandina *et al.* (2001), uses a method that produces an approximate series. Their proposal is considered in this section.

Assume that the usual SSA scheme is performed. A trajectory matrix $(\mathbf{X})$, scatter matrix $(\mathbf{XX}')$, SVD of the scatter $(\mathbf{XX}' = \mathbf{V\Lambda V}')$, matrix of eigenvectors $\left(\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}\right)$ and projection matrix $\left(\mathbf{P}_{\mathfrak{L}_r} = \mathbf{VV}'\right)$ are formed.

The following projected result is produced,

$$\mathbf{Z} = \mathbf{P}_{\mathfrak{L}_r}\mathbf{X}. \tag{3.60}$$

The column vectors of matrix $\mathbf{Z}$ in (3.60) all reside on a single r-flat $\mathfrak{L}_r$. Note that this is not the r-flat H_r. Recall that the column vectors of the trajectory matrix, based on the noise-free signal, reside on the latter r-flat. It is therefore clear that forecasting error will result from this discrepancy, which is due to noise. This fact is inevitable.

The column vectors in (3.60), however, do not have the required Hankel structure. The Hankel structure ensures that reverse diagonal elements are unique. This in turn guarantees a unique folding back to an approximate series. It is therefore a logical step to average over the reverse diagonals of matrix $\mathbf{Z}$, using the Hankelization operation introduced in Chapter 2. Hence the following matrix is formed

$$\tilde{\mathbf{Z}} = \mathcal{H}\mathbf{Z} \,. \tag{3.61}$$

The matrix in (3.61) is then folded into the approximate series $\{f_t^{\sim}\}_{t=1}^{N}$. In this notation the tilde is not supposed to be over-head, so as not to confuse these observations with those belonging to the centred trajectory matrix. When this approximate series is unfolded back into a trajectory matrix, the matrix in (3.61) will result. The column vectors, however, will not reside on a single r-flat. This is due to the Hankelization operation $(\mathcal{H})$ used in (3.61)

.

The approximate series $\{f_t^{\sim}\}_{t=1}^{N}$ can then be applied to (3.39) in Algorithm 3.4 above. Algorithm 3.6 below represents the original algorithm proposed by Golyandina *et al.* (2001).

Algorithm 3.6 Basic SSA R-forecasting due to Golyandina et al. (2001) when no centring is used.

Suppose that a noise-contaminated series $\{x_t\}_{t=1}^{N}$ is considered, with form as described in (3.59).

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $r < \tau \leq n(= N - \tau + 1)$.

(b) The trajectory matrix is not centred and scatter matrix $\mathbf{XX}'$ is formed.

(c) Perform SVD of the scatter matrix $(\mathbf{XX}' = \mathbf{V\Lambda V}')$.

(d) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$.

(e) Form the projection matrix $\mathbf{P}_{\tilde{H}_r} = \mathbf{VV}'$ of the r-flat $\tilde{H}_r$.

(f) Form the projected result $\mathbf{Z} = \mathbf{P}_{\tilde{H}_r}\mathbf{X}$.

(g) Form the approximate series $\{f_t^{\sim}\}_{t=1}^{N}$ by applying the Hankelization operation to the projected result in (f).

(h) Form the vectors $\left\{ \mathbf{v}_j^{\nabla} = \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}' \mathbf{v}_j \middle| j = 1,\dots,r \right\}$.

(i) Form the scalar $\nu^2 = \sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right)^2$ where $\mathbf{e}_\tau$ is the last column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and $\mathbf{v}_j$ is the j^{th} eigenvector derived in step (c).

(j) Calculate the one-period-ahead forecast using

$$\hat{f}_{N+1} = a_1 f_N^{\sim} + a_2 f_{N-1}^{\sim} + \dots + a_{\tau-1} f_{N-\tau+2}^{\sim}, \tag{3.62}$$

where

$$\begin{bmatrix} a_{\tau-1} \\ \vdots \\ a_1 \end{bmatrix} = \frac{1}{1-\nu^2} \sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right) \mathbf{v}_j^{\nabla}. \tag{3.63}$$

No centring of the trajectory matrix is used in Algorithm 3.6 above. In the case of centring, Algorithm 3.7 below can be used.

Algorithm 3.7 Basic SSA R-forecasting due to Golyandina et al. (2001) when centring is used.

Suppose that a noise-contaminated series is considered, with form as described in (3.59).

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $r < \tau \leq n (= N - \tau + 1)$.

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}}$, using the shift vector $\bar{\mathbf{x}} = \frac{1}{n}\mathbf{X1}$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'\right)$.

(e) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_r\end{bmatrix}$.

(f) Form the projection matrix $\mathbf{P}_{\tilde{\mathfrak{L}}_r} = \mathbf{V}\mathbf{V}'$ of the parallel linear subspace $\tilde{\mathfrak{L}}_r$.

(g) Form the projected result $\mathbf{Z} = \bar{\mathbf{x}}\mathbf{1}' + \mathbf{P}_{\tilde{\mathfrak{L}}_r}\tilde{\mathbf{X}}$.

(h) Form the approximate series $\{f_t^{\sim}\}_{t=1}^{N}$ by applying the Hankelization operation to the projected result in (g).

(i) Form the vectors $\left\{\mathbf{v}_j^{\nabla} = \begin{bmatrix}\mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1}\end{bmatrix}' \mathbf{v}_j \,\middle|\, j = 1,\ldots,r\right\}$.

(j) Form the scalar $\nu^2 = \sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)^2$ where $\mathbf{e}_\tau$ is the last column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and $\mathbf{v}_j$ is the j^{th} eigenvector derived in step (c).

(k) Calculate the one-period-ahead forecast using

$$\hat{f}_{N+1} = \bar{x}_\tau + a_1\left(f_N^{\sim} - \bar{x}_{\tau-1}\right) + a_2\left(f_{N-1}^{\sim} - \bar{x}_{\tau-2}\right) + \ldots + a_{\tau-1}\left(f_{N-\tau+2}^{\sim} - \bar{x}_1\right), \tag{3.64}$$

where

$$\begin{bmatrix} a_{\tau-1} \\ \vdots \\ a_1 \end{bmatrix} = \frac{1}{1-\nu^2}\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\mathbf{v}_j^{\nabla}. \tag{3.65}$$

Algorithm 3.8 below illustrates how the generalized algorithm for recurrent one-period-ahead forecasting can be adapted to handle noise-contaminated series.

Algorithm 3.8 Generalized recurrent one-period-ahead forecasting algorithm.

Suppose that a time series $\{x_t\}_{t=1}^{N}$ of the form defined in (3.59) is observed. A recurrent one-period-ahead forecast is constructed using the following steps.

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $r < \tau \leq n (= N - \tau + 1)$.

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}}$, using the shift vector $\overline{\mathbf{x}} = \frac{1}{n}\mathbf{X1}$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V\Lambda V}'\right)$.

(e) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$.

(f) Form the projection matrix $\mathbf{P}_{\tilde{\mathfrak{L}}_r} = \mathbf{VV}'$ of linear subspace $\tilde{\mathfrak{L}}_r$.

(g) Form the projected result $\mathbf{Z} = \overline{\mathbf{x}}\mathbf{1}' + \mathbf{P}_{\tilde{\mathfrak{L}}_r}\tilde{\mathbf{X}}$.

(h) Form the approximate series $\{f_t^{\sim}\}_{t=1}^{N}$ by applying the Hankelization operation to the projected result in (g).

(i) Form the projection matrix $\mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} = \mathbf{I}_\tau - \mathbf{P}_{\tilde{\mathfrak{L}}_r}$ of the orthogonal linear subspace $\mathfrak{L}_r^{\perp}$.

(j) Partition the projection matrix in (g) as follows

$$\begin{aligned} \mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} &= \left[\mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix} \middle| \, \mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} \mathbf{e}_\tau \right] \\ &= \left[\underset{\tau\times(\tau-1)}{\mathbf{P}_1} \middle| \underset{\tau\times 1}{\mathbf{P}_2} \right], \end{aligned} \tag{3.66}$$

where $\mathbf{e}_j$ is the j^{th} column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$.

(k) Calculate the one-period-ahead forecast using

$$\hat{f}_{N+1} = \overline{x}_\tau - \left(\mathbf{P}_2'\mathbf{P}_2\right)^{-1}\left(\mathbf{P}_2'\mathbf{P}_1\right) \begin{bmatrix} f_{N-\tau+2}^{\sim} - \overline{x}_1 \\ \vdots \\ f_N^{\sim} - \overline{x}_{\tau-1} \end{bmatrix}, \tag{3.67}$$

where

$$
\begin{aligned}
\mathbf{P}_1 &= \mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix} \\
\mathbf{P}_2 &= \mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} \mathbf{e}_{\tau} \ .
\end{aligned}
\tag{3.68}
$$

Remark 3.5

To construct an approximate series, use is made of the Hankelization operation. It was mentioned in Chapter 2 that this operation has an undesired property, which was coined the "end-effects". During Hankelization of a matrix, reverse diagonal averaging is performed, cf. steps (g) and (h) in Algorithm 3.8 above. More elements exist for the time series observations closer to the middle of a series during this procedure. Reverse diagonal averaging takes place over a single element for each of the first and last observations of an approximate time series, i.e. for $f_1^{\sim}$ and $f_N^{\sim}$. It would seem that one is faced with a trade-off in this regard. Without reconstruction of an approximate series, forecast errors result due to noise. Forecast errors can now result due to the "end-effects", due to the fact that the last $(\tau - 1)$ approximate series values are used to generate a forecast. More attention will be given to this issue in Section 3.7.

3.3.2 Algorithms using a reconstructed Cadzow-signal

The previous section considered the use of an approximate series during forecasting, as proposed by Golyandina *et al.* (2001). This section proposes the use of a reconstructed signal, using the composite mapping algorithm of Cadzow (1988), which was introduced in Chapter 2. A reconstructed signal using this algorithm will be, for brevity, referred to as a Cadzow-signal.

The main reason for introducing a Cadzow-signal stems from one of the essential requirements of SSA. Column vectors of a trajectory matrix must reside on a single r-flat. The algorithm that Cadzow proposed is iterative by nature and yields a signal series, which conforms to both requirements of SSA. The column vectors of the trajectory matrix, formed using a Cadzow-signal, reside on a single r-flat. When column vectors of the trajectory matrix do not reside on a single r-flat, forecasting error is introduced due to noise.

It is worth taking note of the fact that the approximate series in Section 3.3.1 is also a Cadzow-signal, resulting from a single iteration. The process of projecting and Hankelization, as discussed in Section 3.3.1, is performed until convergence is achieved. This essentially results in a reconstructed Cadzow-signal. The flow diagram in Figure 3.1 below exhibits the process followed in reconstructing a Cadzow-signal.

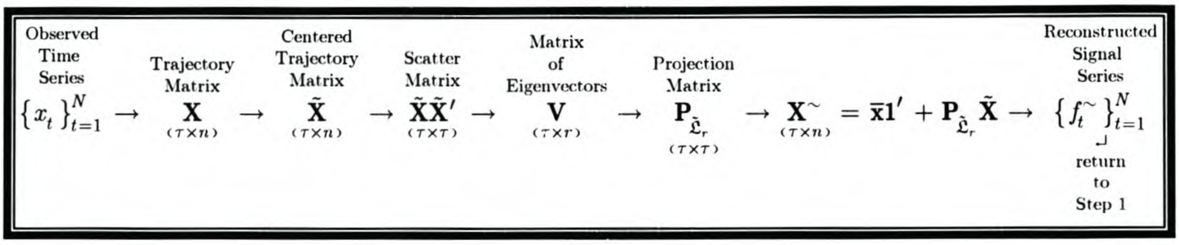

Observed Time Series
$\{x_t\}_{t=1}^{N}$
Trajectory Matrix
$\mathbf{X}$ $(\tau \times n)$
Centered Trajectory Matrix
$\tilde{\mathbf{X}}$ $(\tau \times n)$
Scatter Matrix
$\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$ $(\tau \times \tau)$
Matrix of Eigenvectors
$\mathbf{V}$ $(\tau \times r)$
Projection Matrix
$\mathbf{P}_{\hat{\mathfrak{L}}_r}$ $(\tau \times \tau)$
$\mathbf{X}^{\sim}$ $(\tau \times n)$ $= \bar{\mathbf{x}}\mathbf{1}' + \mathbf{P}_{\hat{\mathfrak{L}}_r}\tilde{\mathbf{X}}$
Reconstructed Signal Series
$\{f_t^{\sim}\}_{t=1}^{N}$
return to Step 1

Figure 3.1 Flow-diagram of Cadzow's iterative property mapping algorithm.

When convergence is achieved, the series $\{f_t^{\sim}\}_{t=1}^{N}$, at the current iteration, is used as the Cadzow-signal. Forecasting can then be performed, using Algorithm 3.9 below.

Algorithm 3.9 Generalized recurrent one-period-ahead forecasting using a Cadzow-signal.

Suppose that a noise-contaminated series, as defined in (3.59), is observed.

(a) Reconstruct a Cadzow-signal $\{f_t^{\sim}\}_{t=}^{N}$ following the procedure outlined in Figure 3.1 above, using known values of the window length (τ) and number of eigenvectors (r).

(b) Use the reconstructed Cadzow-signal in (a) to form the trajectory matrix $(\mathbf{X})$ and shift vector $\bar{\mathbf{x}} = \dfrac{1}{n}\mathbf{X}\mathbf{1}$.

(c) Form the centred trajectory matrix $\tilde{\mathbf{X}}$, using the shift vector.

(d) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(e) Perform SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'\right)$.

(f) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_r\end{bmatrix}$.

(g) Form the projection matrix $\mathbf{P}_{\tilde{\mathfrak{L}}_r} = \mathbf{V}\mathbf{V}'$ of the linear subspace $\tilde{\mathfrak{L}}_r$.

(h) Form the projection matrix $\mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} = \mathbf{I}_\tau - \mathbf{P}_{\tilde{\mathfrak{L}}_r}$ of the orthogonal linear subspace $\tilde{\mathfrak{L}}_r^{\perp}$.

(i) Partition the projection matrix in (h) as follows

$$\begin{aligned} \mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} &= \left[\mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} \left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1} \right] \middle| \, \mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} \mathbf{e}_\tau \right] \\ &= \left[\underset{\tau \times (\tau-1)}{\mathbf{P}_1} \middle| \underset{\tau \times 1}{\mathbf{P}_2} \right], \end{aligned} \tag{3.69}$$

where $\mathbf{e}_j$ is the j^{th} column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$.

(j) Calculate the one-period-ahead forecast using

$$\hat{f}_{N+1} = \overline{x}_\tau - \left(\mathbf{P}_2' \mathbf{P}_2 \right)^{-1} \left(\mathbf{P}_2' \mathbf{P}_1 \right) \begin{bmatrix} f_{N-\tau+2}^{\sim} - \overline{x}_1 \\ \vdots \\ f_N^{\sim} - \overline{x}_{\tau-1} \end{bmatrix}, \tag{3.70}$$

where

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} \left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1} \right] \\ \mathbf{P}_2 &= \mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} \mathbf{e}_\tau \ . \end{aligned} \tag{3.71}$$

Remark 3.6

The reconstructed Cadzow-signal series suffers from the same fate as the approximate series, which was discussed in Remark 3.5 above. The "end-effects" that the Hankelization operation causes, might be slightly more severe in the case of a reconstructed Cadzow-signal. The reason stems from the fact that the Hankelization operation is applied as part of an iterative algorithm. A possible way of overcoming the "end-effects" might be to pursue a vastly different approach to signal reconstruction. It was mentioned in Chapter 2 that a type of STLN (structured total least norms) approach, as proposed by Park *et al*. (1999), might be a viable alternative. This approach is aimed at producing a Hankel-structured matrix with specified rank. The approach is iterative by nature and also uses SVD throughout. The matter was not pursued in this study due to a few reasons. One major reason being the fact that more research in the low-rank Hankel matrix approximation is required. Applied in the SSA context, an arbitrary column vector in the trajectory matrix, based on an observed noisy time series, is used to start the procedure. This in itself is a highly debateable issue.

3.4 K-period-ahead forecasting algorithms for noise-free signals

This section introduces two methods for producing k-period-ahead forecasts directly. The proposed method in Section 3.4.1 below is an extension of the generalized algorithm, which was presented in Section 3.2.3.

3.4.1 Generalized joint-horizon k-period-ahead forecasting algorithm.

This algorithm extrapolates the signal series for a number of periods into the future. It can be likened to a non-parametric method that produces a piecewise smooth curve.

Algorithm 3.10 Generalized joint-horizon k-period-ahead forecasting algorithm.

Suppose that a series $\{f_t\}_{t=1}^{N}$, which is governed by an LRF, is observed.

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $k + r < \tau \leq n (= N - \tau + 1)$.

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}}$, using the shift vector $\bar{\mathbf{x}} = \frac{1}{n}\mathbf{X1}$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V\Lambda V}'\right)$.

(e) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$.

(f) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r} = \mathbf{VV}'$ of the linear subspace $\mathfrak{L}_r$.

(g) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r^{\perp}} = \mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}$ of the orthogonal linear subspace $\mathfrak{L}_r$.

(h) Let k denote the number of forecasts that are required. Partition the projection matrix in (g) as follows

$$\begin{aligned}\mathbf{P}_{\mathfrak{L}_r^{\perp}} &= \left[\mathbf{P}_{\mathfrak{L}_r^{\perp}}\left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-k}\right] \middle| \mathbf{P}_{\mathfrak{L}_r^{\perp}}\left[\mathbf{e}_{\tau-k+1} \quad \cdots \quad \mathbf{e}_{\tau}\right]\right] \\ &= \left[\underset{\tau\times(\tau-k)}{\mathbf{P}_1} \middle| \underset{\tau\times k}{\mathbf{P}_2}\right],\end{aligned} \tag{3.72}$$

where $\mathbf{e}_j$ is the j^{th} column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and k is the number of periods being forecast ahead.

(i) Calculate the joint-horizon k-period-ahead forecasts using

$$\begin{bmatrix}\hat{f}_{N+1} \\ \vdots \\ \hat{f}_{N+k}\end{bmatrix} = \begin{bmatrix}\overline{x}_{\tau-k+1} \\ \vdots \\ \overline{x}_{\tau}\end{bmatrix} - \left(\mathbf{P}_2'\mathbf{P}_2\right)^{-1}\left(\mathbf{P}_2'\mathbf{P}_1\right)\begin{bmatrix}f_{N-\tau+k+1} - \overline{x}_1 \\ \vdots \\ f_N - \overline{x}_{\tau-k}\end{bmatrix}, \tag{3.73}$$

where

$$\begin{aligned}\mathbf{P}_1 &= \mathbf{P}_{\mathfrak{L}_r^{\perp}}\left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-k}\right] \\ \mathbf{P}_2 &= \mathbf{P}_{\mathfrak{L}_r^{\perp}}\left[\mathbf{e}_{\tau-k+1} \quad \cdots \quad \mathbf{e}_{\tau}\right].\end{aligned} \tag{3.74}$$

Remark 3.7

When producing k-period-ahead forecasts using (3.73) the choice of window length must satisfy the condition,

$$k + r < \tau \leq n(= N - \tau + 1). \tag{3.75}$$

Proof.

Suppose that an observed time series $\{f_t\}_{t=1}^{N}$ is governed by an LRF and therefore from the broad class of functions in (3.1). Suppose that the trajectory matrix $(\mathbf{X})$, centred trajectory matrix $(\tilde{\mathbf{X}})$, scatter matrix $(\tilde{\mathbf{X}}\tilde{\mathbf{X}}')$ and SVD of the scatter matrix $(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}')$ were formed. Suppose that r eigenvectors are used to form the matrix

$$\underset{(\tau \times r)}{\mathbf{V}} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}. \tag{3.76}$$

Column vectors of the trajectory matrix $(\mathbf{X})$ reside on a single r-flat H_r, and column vectors of the centred trajectory matrix $(\tilde{\mathbf{X}})$, reside on a single parallel linear subspace $\mathfrak{L}_r$. The projection matrix of the latter subspace is given by

$$\mathbf{P}_{\mathfrak{L}_r} = \mathbf{V}\mathbf{V}'. \tag{3.77}$$

The projection matrix of the orthogonal parallel linear subspace $\mathfrak{L}_r^{\perp}$ is given by

$$\mathbf{P}_{\mathfrak{L}_r^{\perp}} = \mathbf{I}_{\tau} - \mathbf{P}_{\mathfrak{L}_r}. \tag{3.78}$$

The vector,

$$\underset{(\tau \times 1)}{\mathbf{x}_{N+k}} = \begin{bmatrix} f_{N-\tau+k+1} & \cdots & f_N & \hat{f}_{N+1} & \cdots & \hat{f}_{N+k} \end{bmatrix}' \tag{3.79}$$

is formed under the assumption that it resides on r-flat H_r, where $\hat{f}_{N+1}, \ldots, \hat{f}_{N+k}$ are the k-period-ahead forecasts. If the vector in (3.79) resides on r-flat H_r, then the centred vector,

$$\tilde{\mathbf{x}}_{N+k} = \left[\left(f_{N-\tau+k+1} - \overline{x}_1\right) \ \cdots \ \left(f_N - \overline{x}_{\tau-k}\right) \ \left(\hat{f}_{N+1} - \overline{x}_{\tau-k+1}\right) \ \cdots \ \left(\hat{f}_{N+k} - \overline{x}_{\tau}\right) \right]' \tag{3.80}$$

will reside on parallel linear subspace $\mathfrak{L}_\tau$.

It is then clear that,

$$\mathbf{P}_{\mathfrak{L}_\tau^\perp}\tilde{\mathbf{x}}_{N+k} = \mathbf{0}. \tag{3.81}$$

Partition the projection matrix $\mathbf{P}_{\mathfrak{L}_\tau^\perp}$ such that

$$\begin{aligned}\mathbf{P}_{\mathfrak{L}_\tau^\perp} &= \left[\mathbf{P}_{\mathfrak{L}_\tau^\perp}\begin{bmatrix}\mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-k}\end{bmatrix} \,\middle|\, \mathbf{P}_{\mathfrak{L}_\tau^\perp}\begin{bmatrix}\mathbf{e}_{\tau-k+1} & \cdots & \mathbf{e}_\tau\end{bmatrix}\right] \\ &= \left[\underset{\tau\times(\tau-k)}{\mathbf{P}_1} \,\middle|\, \underset{\tau\times k}{\mathbf{P}_2}\right].\end{aligned} \tag{3.82}$$

Combining (3.81) and (3.82) then yields

$$\begin{aligned}\left[\underset{\tau\times(\tau-k)}{\mathbf{P}_1} \,\middle|\, \underset{\tau\times k}{\mathbf{P}_2}\right]\tilde{\mathbf{x}}_{N+k} &= \mathbf{0} \\ \mathbf{P}_2\begin{bmatrix}\hat{f}_{N+1} - \overline{x}_{\tau-k+1} \\ \vdots \\ \hat{f}_{N+k} - \overline{x}_\tau\end{bmatrix} &= -\mathbf{P}_1\begin{bmatrix}f_{N-\tau+k+1} - \overline{x}_1 \\ \vdots \\ f_N - \overline{x}_{\tau-k}\end{bmatrix}.\end{aligned} \tag{3.83}$$

This gives the required result, viz.

$$\begin{bmatrix}\hat{f}_{N+1} \\ \vdots \\ \hat{f}_{N+k}\end{bmatrix} = \begin{bmatrix}\overline{x}_{\tau-k+1} \\ \vdots \\ \overline{x}_\tau\end{bmatrix} - \left(\mathbf{P}_2'\mathbf{P}_2\right)^{-1}\left(\mathbf{P}_2'\mathbf{P}_1\right)\begin{bmatrix}f_{N-\tau+k+1} - \overline{x}_1 \\ \vdots \\ f_N - \overline{x}_{\tau-k}\end{bmatrix} \tag{3.84}$$

and concludes the proof. □

Remark 3.8

The joint-horizon k-period-ahead forecasts exist only if $\left(\mathbf{P}_2'\mathbf{P}_2\right)$ is non-singular. This is the case when the window length satisfies the condition,

$$k + r < \tau \leq n(= N - \tau + 1). \tag{3.85}$$

3.4.2 Recurrent vector k-period-ahead forecasting due to Golyandina *et al.* (2001)

Algorithm 3.11, below, is an algorithmic approach to k-period-ahead vector forecasting in SSA, as proposed in Golyandina *et al.* (2001).

Algorithm 3.11 Vector forecasting due to Golyandina et al. (2001) when no centring is used.

This algorithm is due to Golyandina *et al.* (2001, p.108) and has been modified to produce forecasts for noise-free signals. The notation has also been adapted to suit the purposes of this study.

Suppose that a noise-free signal series, which is governed by an LRF, is observed.

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $r < \tau \leq n(= N - \tau + 1)$.

(b) The trajectory matrix is not centred and scatter matrix $\mathbf{XX}'$ is formed.

(c) Perform SVD of the scatter matrix $(\mathbf{XX}' = \mathbf{V\Lambda V}')$.

(d) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$.

(e) Form the vectors $\left\{ \mathbf{v}_j^{\nabla} = \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}' \mathbf{v}_j \middle| j = 1,\ldots,r \right\}$.

(f) Form the scalar $\nu^2 = \sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right)^2$ where $\mathbf{e}_\tau$ is the last column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and $\mathbf{v}_j$ is the j^{th} eigenvector derived in step (c).

(g) Form the $(\tau - 1) \times (\tau - 1)$ matrix,

$$\mathbf{\Pi} = \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}' \mathbf{V}\mathbf{V}' \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix} + \frac{\left(\sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right) \mathbf{v}_j^{\nabla} \right) \left(\sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right) \mathbf{v}_j^{\nabla} \right)'}{1 - \nu^2}.$$

(3.86)

(h) Form the $\tau \times \tau$ matrix,

$$\underset{(\tau\times\tau)}{\mathbf{P}} = \begin{pmatrix} \mathbf{\Pi} \begin{bmatrix} \mathbf{e}_2 & \cdots & \mathbf{e}_\tau \end{bmatrix}' \\ \frac{1}{1-\nu^2} \left(\sum_{j=1}^{r} \left(\mathbf{e}_\tau' \mathbf{v}_j \right) \mathbf{v}_j^{\nabla} \right)' \begin{bmatrix} \mathbf{e}_2 & \cdots & \mathbf{e}_\tau \end{bmatrix}' \end{pmatrix}. \tag{3.87}$$

(i) Form the following column vectors

$$\underset{(\tau\times 1)}{\mathbf{z}_i} = \begin{cases} \mathbf{x}_i & for \quad i = 1, \ldots, n \\ \mathbf{P}\mathbf{z}_{i-1} & for \quad i = n+1, \ldots, n+k \end{cases}. \tag{3.88}$$

(j) Perform reverse diagonal averaging over the column vectors in (3.88) to produce the series $f_1, \ldots, f_N, \hat{f}_{N+1}, \ldots, \hat{f}_{N+k}$. In this series $\hat{f}_{N+1}, \ldots, \hat{f}_{N+k}$ represent the k-period-ahead forecasts.

Remark 3.9

It is not difficult to show that a more convenient form of (3.86) is

$$\mathbf{\Pi} = \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}' \mathbf{P}_{\mathfrak{L}_r} \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix} + \frac{\begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}' \mathbf{P}_{\mathfrak{L}_r} \mathbf{e}_\tau \mathbf{e}_\tau' \mathbf{P}_{\mathfrak{L}_r} \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}}{\left(1 - \nu^2 \right)},$$

(3.89)

where

$$\mathbf{P}_{\mathfrak{L}_r} = \mathbf{V}\mathbf{V}'. \tag{3.90}$$

Note that (3.87) can then be written as

$$\underset{(\tau\times\tau)}{\mathbf{P}} = \begin{bmatrix} \left\{ \begin{bmatrix} \mathbf{e}_1' \\ \vdots \\ \mathbf{e}_{\tau-1}' \end{bmatrix} \mathbf{P}_{\mathfrak{L}_r} \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix} + \frac{\begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}' \mathbf{P}_{\mathfrak{L}_r} \mathbf{e}_\tau \mathbf{e}_\tau' \mathbf{P}_{\mathfrak{L}_r} \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}}{\left(1-\nu^2\right)} \right\} \begin{bmatrix} \mathbf{e}_2' \\ \vdots \\ \mathbf{e}_\tau' \end{bmatrix} \\ \frac{1}{1-\nu^2} \mathbf{e}_\tau' \mathbf{P}_{\mathfrak{L}_r} \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix} \begin{bmatrix} \mathbf{e}_2 & \cdots & \mathbf{e}_\tau \end{bmatrix}' \end{bmatrix}. \tag{3.91}$$

Note that the projected results, $\left\{ \mathbf{P}\mathbf{z}_{i-1} \middle| i = n+1,\ldots,n+k \right\}$, are constructed under the assumption that $\left\{ \mathbf{P}\mathbf{z}_{i-1} \in H_r \middle| i = n+1,\ldots,n+k \right\}$.

Algorithm 3.12 Vector forecasting due to Golyandina et al. (2001) when centring is used. Suppose that a noise-free signal series, which is governed by an LRF, is observed.

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $r < \tau \leq n (= N - \tau + 1)$.

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}}$, using the shift vector $\bar{\mathbf{x}} = \frac{1}{n}\mathbf{X}\mathbf{1}$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}' \right)$.

(e) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$.

(f) Form the vectors $\left\{ \mathbf{v}_j^{\nabla} = \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}' \mathbf{v}_j \middle| j = 1,\ldots,r \right\}$.

(g) Form the scalar $\nu^2 = \sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)^2$ where $\mathbf{e}_\tau$ is the last column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and $\mathbf{v}_j$ is the j^{th} eigenvector derived in step (c).

(h) Form the $(\tau - 1) \times (\tau - 1)$ matrix

$$\mathbf{\Pi} = \left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1}\right]' \mathbf{V}\mathbf{V}' \left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-1}\right] + \frac{\left(\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\mathbf{v}_j^\nabla\right)\left(\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\mathbf{v}_j^\nabla\right)'}{1-\nu^2}. \tag{3.92}$$

(i) Form the $\tau \times \tau$ matrix

$$\underset{(\tau\times\tau)}{\mathbf{P}} = \begin{pmatrix} \mathbf{\Pi}\left[\mathbf{e}_2 \quad \cdots \quad \mathbf{e}_\tau\right]' \\ \frac{1}{1-\nu^2}\left(\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\mathbf{v}_j^\nabla\right)'\left[\mathbf{e}_2 \quad \cdots \quad \mathbf{e}_\tau\right]' \end{pmatrix}. \tag{3.93}$$

(j) Form the following column vectors

$$\mathbf{z}_i = \begin{cases} \mathbf{x}_i & \mathit{for} \quad i = 1,\ldots,n \\ \overline{\mathbf{x}} + \mathbf{P}\mathbf{z}_{i-1} & \mathit{for} \quad i = n+1,\ldots,n+k. \end{cases} \tag{3.94}$$

(k) Perform reverse diagonal averaging over the column vectors in (3.94) to produce the series $f_1,\ldots,f_N,\hat{f}_{N+1},\ldots,\hat{f}_{N+k}$. In this series $\hat{f}_{N+1},\ldots,\hat{f}_{N+k}$ represent the k-period-ahead forecasts.

3.5 K-period-ahead forecasting algorithms for noise-contaminated signals

The assumption of a noise-free signal, which is governed by an LRF, is relaxed in this section. Two approaches can be followed, viz. using an approximate series or Cadzow-signal series to produce forecasts.

3.5.1 Algorithms using an approximate series

The original algorithm due to Golyandina *et al.* (2001, p.108) that uses vector forecasting is supplied in the form of Algorithm 3.13 below. The algorithm uses projections to produce column vectors that reside on a single r-flat $\tilde{H}_r$. If the Hankelization operation is applied to the resulting projected matrix, an approximate series will result. The column vectors of the projected matrix result are used in ensuing forecasting. The first algorithm does not use centring of the trajectory matrix.

Algorithm 3.13 Vector forecasting due to Golyandina et al. (2001) when no centring is used.

Suppose that a noise-contaminated series $\{x_t\}_{t=1}^{N}$ is available.

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $r < \tau \leq n (= N - \tau + 1)$.

(b) The trajectory matrix is not centred and scatter matrix $\mathbf{XX}'$ is formed.

(c) Perform SVD of the scatter matrix $(\mathbf{XX}' = \mathbf{V\Lambda V}')$.

(d) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$.

(e) Form the projection matrix $\mathbf{P}_{\tilde{H}_r} = \mathbf{VV}'$ of r-flat $\tilde{H}_r$.

(f) Form the projected result $\hat{\mathbf{X}} = \mathbf{P}_{\tilde{H}_r}\mathbf{X}$.

(g) Form the vectors $\left\{ \mathbf{v}_j^{\nabla} = \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1} \end{bmatrix}' \mathbf{v}_j \middle| j = 1,\ldots,r \right\}$.

(h) Form the scalar $\nu^2 = \sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)^2$ where $\mathbf{e}_\tau$ is the last column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and $\mathbf{v}_j$ is the j^{th} eigenvector derived in step (c).

(i) Form the $(\tau - 1)\times(\tau - 1)$ matrix

$$\mathbf{\Pi} = \begin{bmatrix}\mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1}\end{bmatrix}'\mathbf{V}\mathbf{V}'\begin{bmatrix}\mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-1}\end{bmatrix} + \frac{\left(\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\mathbf{v}_j^\nabla\right)\left(\sum_{j=1}^{r}\left(\mathbf{e}_\tau'\mathbf{v}_j\right)\mathbf{v}_j^\nabla\right)'}{1-\nu^2}. \tag{3.95}$$

(j) Form the following column vectors

$$\mathbf{z}_i = \begin{cases}\hat{\mathbf{x}}_i & for \quad i = 1,\ldots,n \\ \mathbf{P}\mathbf{z}_{i-1} & for \quad i = n+1,\ldots,n+k.\end{cases} \tag{3.96}$$

(k) Perform reverse diagonal averaging over the column vectors in (3.96) to produce the series $f_1,\ldots,f_N,\hat{f}_{N+1},\ldots,\hat{f}_{N+k}$. In this series $\hat{f}_{N+1},\ldots,\hat{f}_{N+k}$ represent the k-period-ahead forecasts.

Remark 3.10

The projected results $\left\{\mathbf{P}\mathbf{z}_{i-1}\middle| i = n+1,\ldots,n+k\right\}$ are constructed under the assumption that $\left\{\mathbf{P}\mathbf{z}_{i-1} \in \tilde{H}_r \middle| i = n+1,\ldots,n+k\right\}$. The projected results can be thought of as resulting in the column vectors of the following matrix,

$$\underset{(\tau\times k)}{\mathbf{Z}} = \begin{bmatrix} \mathbf{Px}_n & \mathbf{Pz}_{n+1} & \mathbf{Pz}_{n+2} & \cdots & \mathbf{Pz}_{n+k-1} \end{bmatrix}$$

$$= \begin{bmatrix} \hat{f}_{N-\tau+2} & \hat{f}_{N-\tau+3} & \hat{f}_{N-\tau+4} & \cdots & \hat{f}_{N-\tau+k+1} \\ \hat{f}_{N-\tau+3} & \hat{f}_{N-\tau+4} & \hat{f}_{N-\tau+5} & \cdots & \hat{f}_{N-\tau+k+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{f}_{N-1} & \hat{f}_{N} & \hat{f}_{N+1} & \cdots & \hat{f}_{N+k-2} \\ \hat{f}_{N} & \hat{f}_{N+1} & \hat{f}_{N+2} & \cdots & \hat{f}_{N+k-1} \\ \hat{f}_{N+1} & \hat{f}_{N+2} & \hat{f}_{N+3} & \cdots & \hat{f}_{N+k} \end{bmatrix}. \tag{3.97}$$

The notation used in (3.97) is slightly misleading, as the reverse diagonal elements of this matrix are not unique. This is due to noise in the series. The column vectors that result from the projection reside on a single r-flat $\tilde{H}_r$. The Hankelization operation must therefore be applied to these column vectors, or rather, the matrix $\mathbf{Z}$ in (3.97). This operation will use reverse diagonal averaging and result in the series of forecasts $\hat{f}_{N+1},\dots,\hat{f}_{N+k}$. The vector-forecasts will therefore also suffer the "end-effects" fate that the Hankelization operation carries. It should also be clear that comparisons of forecasting methods, on a theoretical basis, become extremely complex when dealing with noise.

Algorithm 3.13 and Algorithm 3.16 can be consulted for changes that must be made to perform vector forecasting, when centring of the trajectory matrix is used.

Algorithm 3.14 below adapts the generalized k-period-ahead forecasting algorithm. The algorithm makes use of an approximate series to produce forecasts.

Algorithm 3.14 Generalized joint-horizon k-period-ahead forecasting algorithm.

Suppose that a noise-contaminated series, from the broad class as defined in (3.59), is observed.

(a) Unfold the time series into a $\tau \times n$ trajectory matrix $\mathbf{X}$, where $r < \tau \leq n (= N - \tau + 1)$.

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}}$, using the shift vector $\overline{\mathbf{x}} = \dfrac{1}{n}\mathbf{X}\mathbf{1}$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}'\right)$.

(e) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_r\end{bmatrix}$.

(f) Form the projection matrix $\mathbf{P}_{\tilde{\mathfrak{L}}_r} = \mathbf{V}\mathbf{V}'$ of the linear subspace $\tilde{\mathfrak{L}}_r$.

(g) Form the projected result $\mathbf{Z} = \overline{\mathbf{x}}\mathbf{1}' + \mathbf{P}_{\tilde{\mathfrak{L}}_r}\tilde{\mathbf{X}}$.

(h) Form the approximate series $\{f_t^{\sim}\}_{t=1}^{N}$ by applying the Hankelization operation to the projected result in (g).

(i) Let k denote the number of forecasts that are required. Partition the projection matrix in (f) as follows

$$\begin{aligned}\mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}} &= \left[\mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}}\begin{bmatrix}\mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-k}\end{bmatrix} \,\middle|\, \mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}}\begin{bmatrix}\mathbf{e}_{\tau-k+1} & \cdots & \mathbf{e}_{\tau}\end{bmatrix}\right] \\ &= \left[\underset{\tau\times(\tau-k)}{\mathbf{P}_1} \,\middle|\, \underset{\tau\times k}{\mathbf{P}_2}\right],\end{aligned} \tag{3.98}$$

where $\mathbf{e}_j$ is the j^{th} column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and k is the number of periods being forecast ahead.

(j) Calculate the k-period-ahead forecast using

$$\begin{bmatrix}\hat{f}_{N+1} \\ \vdots \\ \hat{f}_{N+k}\end{bmatrix} = \begin{bmatrix}\overline{x}_{\tau-k+1} \\ \vdots \\ \overline{x}_{\tau}\end{bmatrix} - \left(\mathbf{P}_2'\mathbf{P}_2\right)^{-1}\left(\mathbf{P}_2'\mathbf{P}_1\right)\begin{bmatrix}f_{N-\tau+k+1}^{\sim} - \overline{x}_1 \\ \vdots \\ f_N^{\sim} - \overline{x}_{\tau-k}\end{bmatrix}. \tag{3.99}$$

3.5.2 Algorithms that use a reconstructed Cadzow-signal.

Minor changes need to be applied to Algorithm 3.14 above to use a Cadzow-signal during forecasting.

Algorithm 3.15

The first step of this algorithm is to reconstruct a Cadzow-signal. Steps (a) (using the Cadzow-signal) to (j) in Algorithm 3.14 above can then be followed.

This algorithm is the final of the forecasting algorithms considered in the case of single-channel SSA. Section 3.6 will now introduce a forecasting algorithm that can be applied in the multi-channel SSA context. The forecasting algorithm can be used to produce recurrent one-period-ahead or k-period-ahead forecasts.

3.6 General k-period-ahead forecasting algorithm for multi-channel SSA

The following algorithm is supplied without proof. The generalised k-period-ahead forecasting algorithm is extended to apply to the multi-channel SSA case.

Algorithm 3.16 Algorithm for general k-period-ahead forecasts in multi-channel SSA.

Suppose that p time series of common length N are observed. Suppose that each of the series is governed by an LRF.

Denote these time series by $\{f_t^{(1)}\}_{t=1}^N, \{f_t^{(2)}\}_{t=1}^N, \cdots, \{f_t^{(p)}\}_{t=1}^N$, where $f_i^{(j)}$ denotes the i^{th} element of time series j.

The following steps are followed to construct the forecasts.

(a) Unfold each of the time series into a $\tau \times n$ trajectory matrix,

$$\underset{(\tau\times n)}{\mathbf{F}_i} = \begin{bmatrix} f_1^{(i)} & f_2^{(i)} & \cdots & f_n^{(i)} \\ f_2^{(i)} & f_3^{(i)} & \cdots & f_{n+1}^{(i)} \\ \vdots & \vdots & \cdots & \vdots \\ f_\tau^{(i)} & f_{\tau+1}^{(i)} & \cdots & f_N^{(i)} \end{bmatrix} \tag{3.100}$$

for $i = 1,\ldots,p$. Form the $p\tau \times n$ trajectory matrix,

$$\begin{aligned} \underset{(p\tau\times n)}{\mathbf{X}} &= \begin{bmatrix} f_1^{(1)} & f_2^{(1)} & \cdots & f_n^{(1)} \\ f_1^{(2)} & f_2^{(2)} & \cdots & f_n^{(2)} \\ \vdots & \vdots & \cdots & \vdots \\ f_1^{(p)} & f_2^{(p)} & \cdots & f_n^{(p)} \\ \vdots & \vdots & \cdots & \vdots \\ f_\tau^{(1)} & f_{\tau+1}^{(1)} & \cdots & f_N^{(1)} \\ f_\tau^{(2)} & f_{\tau+1}^{(2)} & \cdots & f_N^{(2)} \\ \vdots & \vdots & \cdots & \vdots \\ f_\tau^{(p)} & f_{\tau+1}^{(p)} & \cdots & f_N^{(p)} \end{bmatrix} \\ &= \underset{(\tau\times n)}{\mathbf{F}_1} \otimes \underset{(p\times 1)}{\mathbf{e}_1} + \underset{(\tau\times n)}{\mathbf{F}_2} \otimes \underset{(p\times 1)}{\mathbf{e}_2} + \cdots + \underset{(\tau\times n)}{\mathbf{F}_p} \otimes \underset{(p\times 1)}{\mathbf{e}_p} \end{aligned} \tag{3.101}$$

using the individual trajectory matrices. Form the vector of means,

$$\underset{(p\tau\times 1)}{\bar{\mathbf{x}}} = \frac{1}{n}\mathbf{X}\mathbf{1}_n \tag{3.102}$$

(b) Form the centred trajectory matrix $\tilde{\mathbf{X}}$.

(c) Form the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

(d) Perform SVD of the scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'\right)$.

(e) Select r eigenvectors in matrix $\mathbf{V} = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_r\end{bmatrix}$.

(f) Form the projection matrix $\mathbf{P}_{\tilde{\mathfrak{L}}_r} = \mathbf{V}\mathbf{V}'$ of linear subspace $\tilde{\mathfrak{L}}_r$.

(g) Form the projection matrix $\mathbf{P}_{\tilde{\mathfrak{L}}_r^\perp} = \mathbf{I}_\tau - \mathbf{P}_{\tilde{\mathfrak{L}}_r}$ of the orthogonal linear subspace $\tilde{\mathfrak{L}}_r$.

(h) Let k denote the number of forecasts that are required. Partition the projection matrix in (g) as follows

$$\begin{aligned}\underset{(p\tau\times p\tau)}{\mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}}} &= \left[\mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}}\left\{\mathbf{I}_p\otimes\left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-k}\right]\right\} \,\middle|\, \mathbf{P}_{\tilde{\mathfrak{L}}_r^{\perp}}\left\{\mathbf{I}_p\otimes\left[\mathbf{e}_{\tau-k+1} \quad \cdots \quad \mathbf{e}_{\tau}\right]\right\}\right] \\ &= \left[\underset{p\tau\times(p\tau-pk)}{\mathbf{P}_1} \,\middle|\, \underset{p\tau\times pk}{\mathbf{P}_2}\right].\end{aligned} \tag{3.103}$$

where $\mathbf{e}_j$ is the j^{th} column vector of the $\tau\times\tau$ identity matrix $\mathbf{I}_\tau$, and k is the number of periods being forecast ahead.

(i) Calculate the k-period-ahead forecast using

$$\underset{(pk\times 1)}{\hat{\mathbf{f}}_{N+k}} = \underset{(pk\times 1)}{\overline{\mathbf{x}}_{N+k}} - \left(\mathbf{P}_2'\mathbf{P}_2\right)^{-1}\left(\mathbf{P}_2'\mathbf{P}_1\right)\underset{p(\tau-k)\times 1}{\tilde{\mathbf{f}}_N}, \tag{3.104}$$

where

$$\underset{(pk\times 1)}{\hat{\mathbf{f}}_{N+k}} = \begin{bmatrix}\hat{f}_{N+1}^{(1)} \\ \vdots \\ \hat{f}_{N+1}^{(p)} \\ \hat{f}_{N+2}^{(1)} \\ \vdots \\ \hat{f}_{N+2}^{(p)} \\ \vdots \\ \hat{f}_{N+k}^{(1)} \\ \vdots \\ \hat{f}_{N+k}^{(p)}\end{bmatrix}, \tag{3.105}$$

$$\underset{(pk\times 1)}{\overline{\mathbf{x}}_{N+k}} = \begin{bmatrix} \overline{x}_{p\tau-pk+1} \\ \vdots \\ \overline{x}_{p\tau-pk+p} \\ \overline{x}_{p\tau-pk+p+1} \\ \vdots \\ \overline{x}_{p\tau-pk+2p} \\ \vdots \\ \overline{x}_{p\tau-pk+(k-1)p+1} \\ \vdots \\ \overline{x}_{p\tau} \end{bmatrix} \tag{3.106}$$

and

$$\tilde{\mathbf{f}}_N = \begin{bmatrix} f^{(1)}_{N-\tau+k+1} \\ \vdots \\ f^{(p)}_{N-\tau+k+1} \\ f^{(1)}_{N-\tau+k+2} \\ \vdots \\ f^{(2)}_{N-\tau+k+2} \\ \vdots \\ f^{(1)}_{N} \\ \vdots \\ f^{(p)}_{N} \end{bmatrix} - \begin{bmatrix} \overline{x}_1 \\ \vdots \\ \overline{x}_p \\ \overline{x}_{p+1} \\ \vdots \\ \overline{x}_{2p} \\ \vdots \\ \overline{x}_{p\tau-pk+1} \\ \vdots \\ \overline{x}_{p\tau-pk} \end{bmatrix}. \tag{3.107}$$

Remark 3.11

It is clear from (3.104) that the forecasts exist only if $\left(\mathbf{P}_2'\mathbf{P}_2\right)$ is non-singular.

This concludes the forecasting algorithms, which this study deals with. Section 3.7 will now consider all aspects that influence successful forecasting in SSA.

3.7 Aspects that influence forecasting in SSA

It is important to take note of all the aspects that can cause forecasting in SSA to fail. Some of these aspects were briefly mentioned in this Chapter, when forecasting algorithms were introduced. Thorough consideration is given in this section to all issues and other problems that can affect successful forecasting in SSA.

3.7.1 Verticality condition

It was clear from the formulation due to Venter (1998), Algorithm 3.2 in Section 3.2, that the spanning vectors of a linear subspace cannot include a vertical space. This implied that

$$\mathbf{e}_\tau \notin \mathfrak{L}_r, \tag{3.108}$$

where $\mathfrak{L}_r = span(\mathbf{v}_1,\dots,\mathbf{v}_r)$, and the vectors $\mathbf{v}_1,\dots,\mathbf{v}_r$ represent the eigenvectors which are formed during SVD of the scatter matrix. The condition in (3.108) is termed the *verticality condition* (Golyandina *et al.*, 2001). The condition must be satisfied before any of the forecasting algorithms, presented in this Chapter, can be applied.

Recall that the recurrent one-period-ahead forecasting formulation due to Venter (1998) was of the form

$$\hat{f}_{N+1} = \overline{x}_\tau + \frac{\left\langle \mathbf{P}_{\mathfrak{L}_r} \tilde{\mathbf{x}}_{n+1}^{(0,\tau)}, \mathbf{e}_\tau \right\rangle}{1 - \left\| \mathbf{P}_{\mathfrak{L}_r} \mathbf{e}_\tau \right\|^2}. \tag{3.109}$$

It is clear from this that large forecasting errors will occur when

$$\left\| \mathbf{P}_{\mathfrak{L}_r} \mathbf{e}_\tau \right\| \approx 1. \tag{3.110}$$

This demands a stronger condition than (3.108), viz. that $\left\| \mathbf{P}_{\mathfrak{L}_r} \mathbf{e}_\tau \right\|$ must not be close to one. Let us now consider the latter in detail. (3.110) can be thought of as the angle between the linear subspace $\mathfrak{L}_r$ and the vertical space, spanned by $\mathbf{e}_\tau$, being close to zero. It is thus clear that the angle between each of the spanning vectors $\mathbf{v}_1,\ldots,\mathbf{v}_r$ and $\mathbf{e}_\tau$ should not be close to zero.

It is a standard result (Wickens, 1995) that the cosine of the angle between two, n-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$, is given by

$$\cos\angle(\mathbf{x},\mathbf{y}) = \frac{\langle \mathbf{x},\mathbf{y}\rangle}{\|\mathbf{x}\|\|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} y_i^2\right)}}. \tag{3.111}$$

Suppose that the column vectors of matrix $\mathbf{V}$ contain the first r eigenvectors, obtained through SVD of the trajectory matrix. The cosine of the angle between $\mathbf{v}_j$ and $\mathbf{e}_\tau = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}'$ is then given by,

$$\begin{aligned} \cos\angle\left(\mathbf{v}_j,\mathbf{e}_\tau\right) &= \frac{\left\langle \mathbf{v}_j,\mathbf{e}_\tau \right\rangle}{\left\|\mathbf{v}_j\right\|\|\mathbf{e}_\tau\|} \\ &= \sum_{i=1}^{\tau} v_{i,j} e_i \\ &= v_{\tau,j} \end{aligned} \tag{3.112}$$

for $j = 1,\ldots,r$, where

$$\begin{aligned} v_{ij} &= (i,j)^{\text{th}} \text{ element of matrix } \mathbf{V} \\ e_i &= i^{th} \text{ element of } \mathbf{e}_\tau \\ \|\mathbf{e}_\tau\| &= 1 \ (\text{vector has unit length}) \\ \left\|\mathbf{v}_j\right\| &= 1 \ \left(j^{th} \text{ eigenvector has unit length}\right). \end{aligned}$$

If the angle between the j^{th} eigenvector ($\mathbf{v}_j$) and vertical space spanned by ($\mathbf{e}_\tau$) must not be close to zero, then this translates to the cosine of the angle not being too close to 1.

Hence, for (3.110) not to hold, we require

$$v_{\tau,j} \ll 1, \; j = 1,\dots,r \tag{3.113}$$

where $\left\{ v_{\tau,j} \middle| j = 1,\dots,r \right\}$ are the $(\tau, j)^{th}$ elements of the matrix of eigenvectors $(\mathbf{V})$.

Any software package that performs SSA forecasting must therefore display the verticality check, presented in (3.113). If the cosine of the angle between spanning vectors of the linear subspace, and vertical space, gets too close to 1, extreme caution should be used before producing forecasts.

3.7.2 Separability

Another problem that can be encountered in SSA forecasting, is that of separability. Golyandina *et al.* (2001) devote a substantial amount of effort to this topic. Separability can be considered as the problem of extracting a signal from a noise-distorted series, which has the form

$$x_t = f_t^{(1)} + f_t^{(2)}, \; t = 1,\dots,N\,, \tag{3.114}$$

where $\{f_t^{(1)}\}_{t=1}^{N}$ is a noise-free signal series and $\{f_t^{(2)}\}_{t=1}^{N}$ a noise series.

Separability can, alternatively, be viewed as extracting a specific signal component (such as trend or a cyclical pattern) from a signal with the form in (3.114). In this scenario, $\{f_t^{(1)}\}_{t=1}^{N}$ in (3.114) might be a signal series that contains trend or cyclical behaviour, which must be separated from series $\{f_t^{(2)}\}_{t=1}^{N}$, which might also be a signal series that contains trend or a cyclical pattern. The reason for the latter type of extraction stems from the situation where only a certain informative part of a time series, such as trend, must be either reconstructed or forecast. Choice of the window length (τ) is, according to Golyandina *et al.* (2001), an important consideration that must be taken into account. If the window length is too small then separability of signals becomes difficult and erroneous forecasting results.

The situation is illustrated by Example 3.1 below.

Example 3.1

The following example serves as a counter example to Example 2.2, entertained in Golyandina *et al.* (2001, p.105). Their example is used to illustrate the effect of window length (τ) on separability, as well as the effect on recurrent one-period-ahead forecasting.

The example uses the following time series

$$x_t = f_t^{(1)} + f_t^{(2)},\ t = 1,\ldots,100, \tag{3.115}$$

where

$$\begin{aligned} f_t^{(1)} &= 3(1.01)^{t-1} \\ f_t^{(2)} &= \sin(2\pi(t-1)/10). \end{aligned} \tag{3.116}$$

The series $\{f_t^{(1)}\}_{t=1}^{N}$ is governed by an LRF, which has a trajectory matrix of rank 1, whilst the series $\{f_t^{(2)}\}_{t=1}^{100}$ is governed by an LRF, which has a trajectory matrix of rank 2.

Window lengths of $\tau = 15$ and $\tau = 50$ were applied. The trajectory matrix was constructed in each of the latter cases, using no centring and also using centring. Only the leading eigenvector $\mathbf{v}_1$ was employed in each instance. The leading eigenvector is supposed to represent the pattern described by series $\{f_t^{(1)}\}_{t=1}^{100}$, since the maximum amount of variation in the series $\{x_t\}_{t=1}^{100}$ is due to it. According to Golyandina *et al.* (2001, p.105) the choice of $\tau = 15$ is too low to obtain a good separation of the series. This will consequently produce erroneous forecasts.

Ninety recurrent one-period-ahead forecasts were generated starting at time $t = 101$, i.e. $\hat{f}_{101},\ldots,\hat{f}_{190}$. Three series were used to generate forecasts with: (i) the original series $\{x_t\}_{t=1}^{100}$, (ii) the approximate series and (iii) reconstructed Cadzow-signal series. The results are summarised in Tables 3.1 to 3.4 below. Two measures of forecasting errors were calculated for the forecasts, viz.

$$\begin{aligned} MAPE &= \frac{1}{90}\sum_{t=101}^{190}\left|f_t^{(1)} - \hat{f}_t\right| / f_t^{(1)} \times 100\% \\ MSE &= \frac{1}{90}\sum_{t=101}^{190}\left(f_t^{(1)} - \hat{f}_t\right)^2, \end{aligned} \tag{3.117}$$

where

$$f_t^{(1)} = 3(1.01)^{t-1},\ t = 101,\ldots,190. \tag{3.118}$$

It is clear from Table 3.1 and Table 3.3 that a larger window length results in better forecasting results. According to the MSE measure of forecasting error, there seems to be far less bias, irrespective of whether the original or reconstructed series is used to forecast with. A Cadzow-signal was also utilised in this example. It is clear that the Cadzow-signal is rather robust, irrespective of the window length used. Furthermore, use of the Cadzow-signal has outperformed all the series used to forecast with. The Cadzow-signal has the lowest MSE throughout.

This example therefore shows that the use of a Cadzow-signal has definite advantages. Even when the true underlying window length is not known, the Cadzow-signal will produce better results.

It would therefore seem that use of a Cadzow-signal, can overcome the problem of separability due to a bad choice of window length. The Cadzow-signal does not seem sensitive to the choice of window length, and will produce a signal with desired properties, as already discussed.

Table 3.1 Recurrent one-period-ahead forecasting results for a window length $\tau = 15$ when no centring is used.

	Measures of forecasting accuracy	
Series used	**MAPE**	**MSE**
Original series	7.83%	1.68674
Approximate series	6.71%	1.20953
Cadzow-signal	**5.05%**	**0.49932**

Table 3.2 Recurrent one-period-ahead forecasting results for a window length $\tau = 15$ when centring is used.

	Measures of forecasting accuracy	
Series used	**MAPE**	**MSE**
Original series	8.50%	2.01570
Approximate series	7.25%	1.45842
Cadzow-signal	**5.19%**	**0.52362**

Table 3.3 Recurrent one-period-ahead forecasting results for a window length $\tau = 50$ when no centring is used.

	Measures of forecasting accuracy	
Series used	**MAPE**	**MSE**
Original series	5.30%	0.57995
Approximate series	5.16%	0.52566
Cadzow-signal	**5.03%**	**0.49951**

Table 3.4 Recurrent one-period-ahead forecasting results for a window length $\tau = 50$ when centring is used.

	Measures of forecasting accuracy	
Series used	**MAPE**	**MSE**
Original series	19.51%	11.94394
Approximate series	19.04%	11.60049
Cadzow-signal	**5.24%**	**0.53617**

3.7.3 End-effect of Hankelization operation

When faced with a noise-contaminated series, many considerations must be taken into account. The column vectors of a trajectory matrix, formed using such a series, do not reside on a single r-flat. Forecasting errors will result in such a situation because of noise. If recurrent one-period-ahead forecasting is used to construct a number of forecasts, cumulative errors will be built into the scheme. The errors will mainly be due to bias, which is a direct result of noise. Each consecutive forecast that is produced by the recurrent scheme, will inherit the forecasting error in the form of bias that accumulated to that point.

Golyandina *et al*. (2001) proposed the use of an approximate series $\{f_t^{\sim}\}_{t=1}^{N}$, when faced with the situation of an observed noise-contaminated series. The approximate series is obtained through a projection of the original trajectory matrix onto an r-flat, followed by a Hankelization operation. Suppose that the projected result was of the form

$$\underset{(\tau\times n)}{\mathbf{Z}} = \mathbf{P}_{\tilde{H}_r}\mathbf{X} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ z_{\tau 1} & z_{\tau 2} & \cdots & z_{\tau n} \end{bmatrix}. \tag{3.119}$$

The Hankelization operation is a linear operation. It essentially involves reverse diagonal averaging of the matrix presented in (3.119). The operation is performed using

$$f_s^{\sim} = \begin{cases} \dfrac{1}{s}\displaystyle\sum_{i=1}^{s} z_{i,s-i+1} & for \quad 1 \leq s \leq \tau, \\ \dfrac{1}{\tau}\displaystyle\sum_{i=1}^{\tau} z_{i,s-i+1} & for \quad \tau \leq s \leq n, \\ \dfrac{1}{N-s+1}\displaystyle\sum_{i=1}^{N-s+1} z_{i+s-n,n-i+1} & for \quad n \leq s \leq N. \end{cases} \tag{3.120}$$

It is clear from (3.119) and (3.120) that fewer elements are averaged over during this operation, when producing observations towards both ends of the approximate series $\{f_t^{\sim}\}_{t=1}^{N}$. This raises a valid point of concern in the forecasting paradigm. All the forecasting algorithms entertained in this Chapter are dependent on the last $(\tau - 1)$ observations of the series used in the forecasting scheme. It is not possible to analytically trace the forecasting error that will result due to the "end effects" problem.

Reconstruction of a Cadzow-signal series is iterative by nature. The procedure also uses the Hankelization operation and therefore suffers from the same fate, as was the case of the approximate series. The Cadzow-signal, however, might be slightly worse off because of the "end-effects" being "magnified" due to the iterative scheme. The numeric nature of both schemes, make it analytically impossible to determine errors that might arise from this situation.

Both the above approaches, i.e. using an approximate series or a Cadzow-signal series, try to resolve problems caused by noise. The SSA scheme requires a low-rank Hankel structured trajectory matrix, with column vectors that reside on a single r-flat. Both the mentioned series address this issue, but inherit the "end effects" from the Hankelization operation. It is not envisaged that this problem will be overcome easily. The problem does not seem to affect forecasts much, but is a theoretical glitch in the forecasting scheme that must be pointed out.

As mentioned earlier in this Chapter, a vastly different approach to the above solutions might be required. An approximate series, formed using a reconstruction of a low-rank

Hankel structured matrix, might provide a solution to this issue in future. The approach was not pursued due to reasons mentioned at the end of Section 3.3.

3.7.4 Outliers

The last $(\tau - 1)$ observations of a time series are used to produce a recurrent one-period-ahead forecast. It is therefore clear that the presence of an outlier in this section of the series will result in forecasting errors. If an outlier is present in any other position in the series, it will cause bias in the forecasts. This is due to the well-documented fact that SVD is sensitive to outliers.

Suppose that a single additive outlier is present in a time series, viz.

$$x_t = f_t + a \text{ for some } t \in \{1, \ldots, N - \tau + 1\} \tag{3.121}$$

where a represents the outlier. The SSA scheme produces a multivariate framework and the single outlier will invite itself into τ consecutive column vectors of the trajectory matrix. Detection of such an outlier will be a difficult task in the presence of noise, because of the inherent masking effect. These issues will be addressed in Chapter 4 when outlier identification is considered.

It can be mentioned that it has been found, in this study, that a reconstructed Cadzow-signal is extremely robust against the presence of outliers.

3.7.5 Signal-to-noise ratio

It is obvious that signal-to-noise ratio is important in the SSA scheme. If a signal series is severely distorted by noise, then SVD of the trajectory matrix will suffer from it. The true signal-to-noise ratio will, however, never be known in practise since the true underlying signal is seldom, if ever, known.

3.8 Examples

A few well-known time series are considered in this section for illustrative purposes. The time series that are considered have patterns that fit within the framework that SSA can handle. The window length (τ) and number of eigenvectors (r) used in all the series are arbitrary. Chapter 4 will address methods for making proper choices in this regard.

3.8.1 The Airline Passenger Time Series

The well-known international airline passenger series is considered in this example. The series is well researched and appears in popular time series literature (Box and Jenkins, 1970; Bowerman and O'Connell, 1993; Harvey, 1993; Chatfield, 1994). A more recent article (Farraway and Chatfield, 1998) used the series in a neural network setting. The series consists of thousands of airline passengers that travelled monthly during the period January 1949 to December 1960. The series is exhibited in Figure 3.2 below.

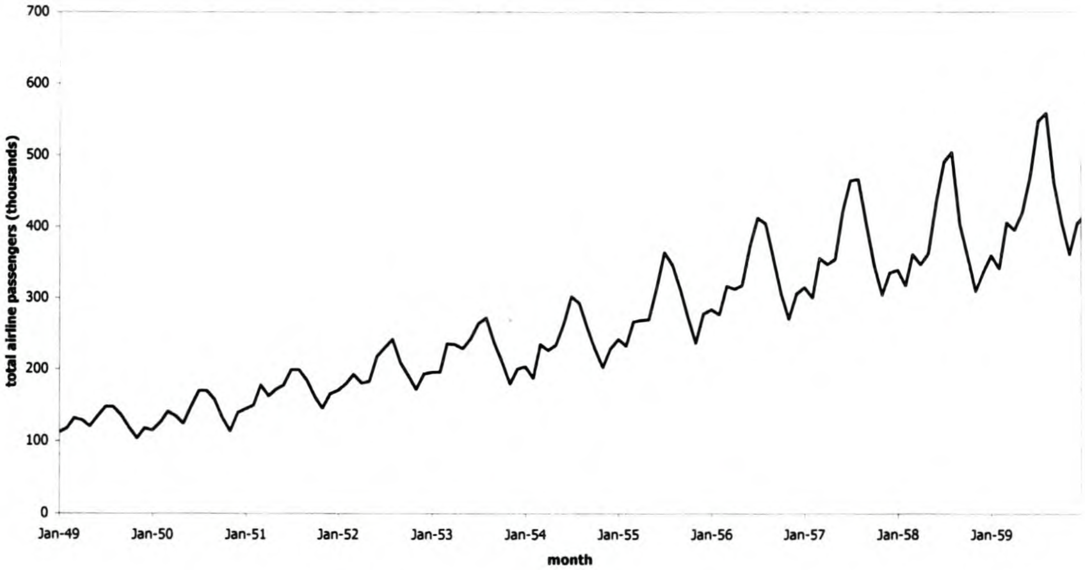


Figure 3.2 Monthly international airline passengers (in thousands).

From perusal of Figure 3.2, it is clear that this series contains a trend and seasonal variation. The series also contains heteroskedasticity, and is customarily log-transformed.

The series was split into two parts for purposes of this example. The first section $\{x_t\}_{t=1}^{132}$, which constitutes monthly observations for 11 years, was used for model fit. The second section $\{x_t\}_{t=133}^{144}$, which represents a full year's monthly observations, was used for testing forecasting accuracy.

Three measures of forecasting accuracy were calculated, viz.

$$\begin{aligned} MAD &= \frac{1}{12}\sum_{t=133}^{144}|x_t - \hat{x}_t| \\ MAPE &= \frac{1}{12}\sum_{t=133}^{144}|x_t - \hat{x}_t| / x_t \times 100\% \\ MSE &= \frac{1}{12}\sum_{t=133}^{144}(x_t - \hat{x}_t)^2. \end{aligned} \tag{3.122}$$

Each of the above measures of forecasting accuracy is reported.

The training series, of length $N = 132$, dictated use of a window length $\tau = 66$. In all the results that follow, the 5 leading eigenvectors were used. Tables 3.5 to 3.8 represent results obtained through the use of recurrent one-period-ahead forecasting. A log-transformation was also used, where indicated.

The results summarised in Tables 3.9 to 3.12, were obtained through use of the generalised joint horizon k-period-ahead forecasting algorithm in Algorithm 3.14.

Different combinations were investigated, which employed the (i) original series, (ii) approximate and (iii) Cadzow-signal series. The vector forecasting results, which accompany each of the latter tables, were obtained from using the CaterpillarSSA software package.

Table 3.5 Recurrent one-period-ahead forecasting results when no transformation, window length $\tau = 66$, no centring of trajectory matrix and the 5 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	22.28704	4.74%	729.00379
Approximate series	25.78241	5.42%	927.71653
Cadzow-signal	37.97684	7.91%	1825.62097

Table 3.6 Recurrent one-period-ahead forecasting results when no transformation, window length $\tau = 66$, centring of trajectory matrix and the 5 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	16.71854	3.67%	425.97329
Approximate series	17.33036	3.80%	460.20553
Cadzow-signal	20.89714	4.38%	602.26714

Table 3.7 Recurrent one-period-ahead forecasting results when a log-transformation, window length $\tau = 66$, no centring of trajectory matrix and the 5 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	40.86665	8.53%	2090.35399
Approximate series	49.18261	10.23%	3004.06294
Cadzow-signal	72.56101	15.12%	6079.26704

Table 3.8 Recurrent one-period-ahead forecasting results when a log-transformation, window length $\tau = 66$, centring of trajectory matrix and the 5 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	14.54298	3.25%	315.12409
Approximate series	14.53482	3.25%	317.21293
Cadzow-signal	17.58471	3.80%	417.48085

Table 3.9 k-period-ahead forecasting results when no transformation, window length $\tau = 66$, no centring of trajectory matrix and the 5 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	21.16128	4.43%	656.59892
Approximate series	23.68480	4.92%	797.47906
Cadzow-signal	37.97985	7.91%	1825.89700
Vector forecasting	25.95172	5.55%	905.54783

Table 3.10 k-period-ahead forecasting results when no transformation, window length $\tau = 66$, centring of trajectory matrix and the 5 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	14.65720	3.16%	344.40370
Approximate series	15.55250	3.34%	372.75884
Cadzow-signal	20.89899	4.38%	602.43619
Vector forecasting	16.15309	3.58%	413.36835

Table 3.11 k-period-ahead forecasting results when a log-transformation, window length $\tau = 66$***, no centring of trajectory matrix and the*** 5 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	40.65621	8.45%	2071.94827
Approximate series	47.73652	9.89%	2839.83175
Cadzow-signal	72.57531	15.12%	6081.69027
Vector forecasting	56.04110	11.75%	3678.56618

Table 3.12 k-period-ahead forecasting results when a log-transformation, window length $\tau = 66$***, centring of trajectory matrix and the*** 5 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	16.25101	3.46%	330.48020
Approximate series	16.00957	3.42%	309.77954
Cadzow-signal	17.58711	3.80%	417.60137
Vector forecasting	14.32862	3.21%	312.14818

A few interesting facts can be deduced from perusal of the above tables. It would seem that centring of the trajectory matrix produces superior results, compared to that of no centring. It is also evident that the use of a log-transformation produced better overall results, when centring was used.

It is, however, too early to reach conclusions. The choice of window length and number of eigenvectors used, were arbitrary. Comparisons will be more valid when model selection is performed, prior to such comparisons.

It would also seem that forecasting results due to use of the original series, at times, produces better results. It should be pointed out that the use of the original series is not entirely justified from a theoretical viewpoint. The original series contains noise and the column vectors of its trajectory matrix do not lie on a single r-flat.

3.8.2 The Hotel Time Series

Another popular time series is the hotel time series. The series considered in this example consists of monthly hotel room averages, occupied during the period January 1977 to December 1991 (Bowermann and O'Connel, 1993; p.523). The series is exhibited in Figure 3.3 below.

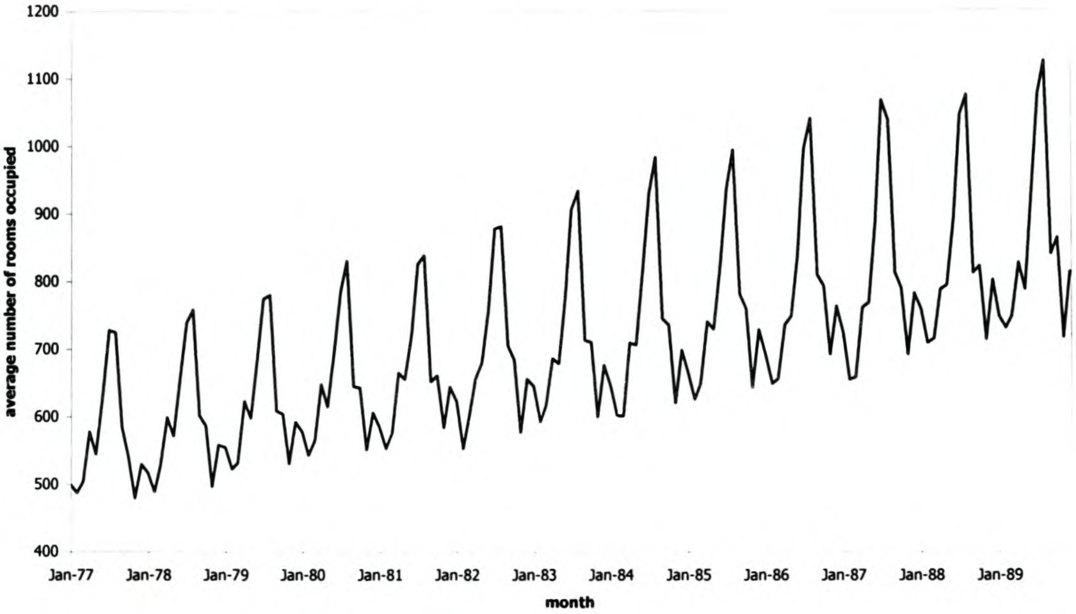


Figure 3.3 Average monthly hotel room occupancy.

This series contains trend and monthly cyclical pattern and no apparent heteroskedasticity. A log-transform is therefore not required.

The series was split into two parts for purposes of this example. The first section $\{x_t\}_{t=1}^{156}$, which constitutes monthly observations for 13 years, was used for model fit. The second section $\{x_t\}_{t=157}^{168}$, which represents a full year's monthly observations, was used for testing forecasting accuracy.

Three measures of forecasting accuracy were calculated, viz.

$$\begin{aligned} MAD &= \frac{1}{12}\sum_{t=157}^{168} \left| x_t - \hat{x}_t \right| \\ MAPE &= \frac{1}{12}\sum_{t=157}^{168} \left| x_t - \hat{x}_t \right| / x_t \times 100\% \\ MSE &= \frac{1}{12}\sum_{t=157}^{168} \left(x_t - \hat{x}_t \right)^2. \end{aligned} \tag{3.123}$$

The training series, of length $N = 156$, dictated use of a window length $\tau = 78$. In all the results that follow, the 12 leading eigenvectors were used. Tables 3.13 to 3.14 represent results obtained through the use of recurrent one-period-ahead forecasting.

The results summarised in Tables 3.15 to 3.16, were obtained through use of the generalised k-period-ahead forecasting algorithms. Different combinations were investigated, which employed the (i) original series, (ii) approximate and (iii) Cadzow-signal series. Vector forecasting results also accompany each of the latter tables at the bottom.

Table 3.13 Recurrent one-period-ahead forecasting results when no transformation, window length $\tau = 78$***, no centring of trajectory matrix and the*** 12 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	19.46673	2.12%	564.12708
Approximate series	20.50461	2.21%	667.82065
Cadzow-signal	24.30687	2.62%	940.72044

Table 3.14 Recurrent one-period-ahead forecasting results when no transformation, window length $\tau = 78$, centring of trajectory matrix and the 12 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	19.13791	2.16%	473.97503
Approximate series	20.05947	2.24%	515.41662
Cadzow-signal	21.93075	2.43%	645.96879

Table 3.15 k-period-ahead forecasting results when no transformation, window length $\tau = 78$, no centring of trajectory matrix and the 12 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	19.17448	2.10%	527.12377
Approximate series	20.27452	2.18%	648.63429
Cadzow-signal	24.30512	2.62%	940.61656
Vector forecasting	21.07364	2.27%	752.90371

Table 3.16 k-period-ahead forecasting results when no transformation, window length $\tau = 78$, centring of trajectory matrix and the 12 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	18.43912	2.05%	460.51402
Approximate series	20.29202	2.25%	531.91952
Cadzow-signal	21.93052	2.43%	645.98906
Vector forecasting	20.54366	2.31%	566.18518

From perusal of the above tables it is clear that forecasts generated using the Cadzow-signal series does not always compare favourably with other methods.

3.8.3 The Monthly Liquor Sales Time Series

The final example of this Chapter uses a series consisting of monthly U.S. liquor sales (Diebold, 1998, p.222) over the period January 1967 to December 1994. The time series is exhibited in Figure 3.4 below.

The series in Figure 3.4 exhibits trend and a cyclical movement. It is also clear that severe heteroscedasticity is present and a log-transformation should be applied.

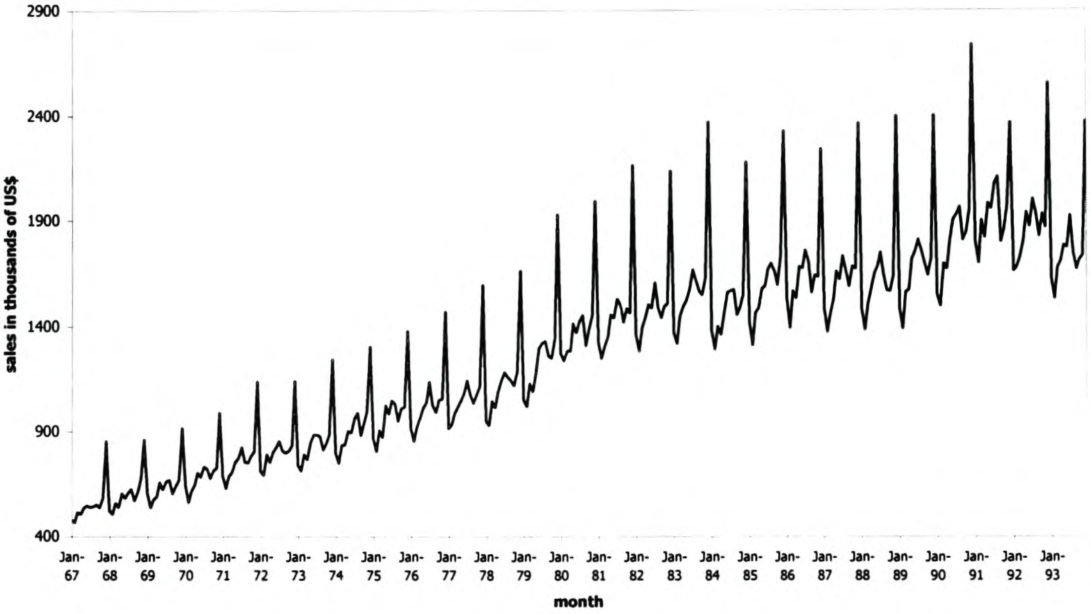


Figure 3.4 Monthly sales of liquor in the U.S. (thousands of $US)

The series was split into two parts for purposes of this example. The first section $\{x_t\}_{t=1}^{324}$, which constitutes monthly observations for 27 years, was used for model fit. The second section $\{x_t\}_{t=325}^{336}$, which represents a full year's monthly observations, was used for testing forecasting accuracy.

The usual measures of forecasting accuracy were calculated, viz.

$$\begin{aligned} MAD &= \frac{1}{12}\sum_{t=325}^{326} |x_t - \hat{x}_t| \\ MAPE &= \frac{1}{12}\sum_{t=325}^{326} |x_t - \hat{x}_t| / x_t \times 100\% \\ MSE &= \frac{1}{12}\sum_{t=325}^{326} (x_t - \hat{x}_t)^2. \end{aligned} \tag{3.124}$$

The training series, of length $N = 324$, dictated use of a window length $\tau = 162$. In all the results that follow, the 12 leading eigenvectors were used.

Tables 3.17 to 3.20 represent results obtained through the use of recurrent one-period-ahead forecasting. Log-transformation was also used, where indicated.

The results summarised in Tables 3.21 to 3.24 were obtained through use of the generalised k-period-ahead forecasting algorithms. Different combinations were investigated, which employed the (i) original series, (ii) approximate and (iii) Cadzow-signal series. Vector forecasting results also accompany each of the latter tables at the bottom.

Table 3.17 Recurrent one-period-ahead forecasting results when no transformation, window length $\tau = 162$, ***no centring of trajectory matrix and the*** 12 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	119.27405	6.79%	17743.51710
Approximate series	121.64792	6.86%	19728.41330
Cadzow-signal	84.65311	4.74%	9865.41805

Table 3.18 Recurrent one-period-ahead forecasting results when no transformation, window length $\tau = 162$, centring of trajectory matrix and the 12 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	153.87767	8.78%	25665.87355
Approximate series	147.60059	8.30%	25150.75291
Cadzow-signal	240.91454	13.45%	66112.38809

Table 3.19 Recurrent one-period-ahead forecasting results when log-transformation, window length $\tau = 162$, no centring of trajectory matrix and the 12 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	90.21436	5.29%	10382.85412
Approximate series	82.05337	4.82%	9209.59857
Cadzow-signal	41.77957	2.45%	3174.19253

Table 3.20 Recurrent one-period-ahead forecasting results when log-transformation, window length $\tau = 162$, centring of trajectory matrix and the 12 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	153.52578	8.93%	25912.91566
Approximate series	155.78495	9.00%	26909.14050
Cadzow-signal	76.37325	4.45%	8103.26077

Table 3.21 k-period-ahead forecasting results when no transformation, window length $\tau = 162$***, no centring of trajectory matrix and the*** 12 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	135.42802	7.75%	21728.30074
Approximate series	136.43216	7.73%	22835.28211
Cadzow-signal	84.62808	4.74%	9859.04209
Vector forecasting	150.67337	8.60%	28106.72624

Table 3.22 k-period-ahead forecasting results when no transformation, window length $\tau = 162$***, centring of trajectory matrix and the*** 12 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	157.52984	9.00%	26752.41578
Approximate series	160.12051	9.05%	28694.40856
Cadzow-signal	240.92157	13.45%	66115.40170
Vector forecasting	257.16370	14.64%	70109.41232

Table 3.23 k-period-ahead forecasting results when a log-transformation, window length $\tau = 162$***, no centring of trajectory matrix and the*** 12 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	101.05843	5.94%	12680.83576
Approximate series	89.32872	5.26%	10545.99147
Cadzow-signal	41.78270	2.44%	3178.36610
Vector forecasting	116.84557	6.83%	16690.18882

Table 3.24 k-period-ahead forecasting results when a log-transformation, window length $\tau = 162$, centring of trajectory matrix and the 12 leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	157.29066	9.15%	27112.63526
Approximate series	165.00742	9.57%	29715.26160
Cadzow-signal	76.35666	4.45%	8099.64892
Vector forecasting	273.14895	15.66%	78087.74762

From perusal of the above tables it becomes evident that the forecasts generated using the Cadzow-signal, when using a log-transformation, produced superior results to the other methods.

3.9 Conclusions and Recommendations

This chapter considered all the recurrent one-period-ahead forecasting algorithms that are available in SSA (Danilov, 1997; Venter, 1998, Golyandina *et al*., 2001). A new alternative coordinate-free formulation to one-period-ahead forecasting was derived, and generalized to produce joint-horizon k-period-ahead forecasts. This result also led to the stabilization of model selection in Chapter 4, which follows. Algorithm 3.14 is the main contribution of this Chapter. This algorithmic application of joint-horizon k-period-ahead forecasting can also handle recurrent one-period-ahead forecasting, as a special case. The approach is totally vectorised and would therefore suit deployment in packages such as MATLAB.

The joint-horizon k-period-ahead forecasting algorithm was also extended to the multi-channel SSA case. A literature study revealed that Algorithm 3.16 is probably one of the first available forecasting algorithms in multi-channel SSA.

Another benefit of this Chapter, in general, is the coherent way in which algorithms are provided. The algorithms would make it possible to incorporate SSA forecasting into commercial software packages in the near future.

Section 3.3 proposed two methods of dealing with noise-contaminated series. The reconstruction of an approximate series and the reconstruction of a Cadzow-signal series were mainly considered. The latter technique is new to SSA and is an advance due to the research of this thesis. Section 3.3.2 also recommended further research in the context of signal reconstruction. It was recommended that the STLN (structured total least norms) approach be investigated for signal reconstruction purposes. Possible pitfalls resulting due to the use of such a technique were also outlined.

The reconstruction of a Cadzow-signal also provides additional benefits to analysis in SSA. Outlier identification becomes possible and is discussed in Chapter 4.

Aspects that can have an influence on successful forecasting in SSA were also elaborated upon in Section 3.7. Topics that were discussed included the verticality condition, "end-effects" of the Hankelization operation, separability of functions, the presence of outliers and the signal-to-noise ratio.

Section 3.8 concluded with a few convincing examples of the effectiveness of forecasting in SSA. It was evident from the examples in this Chapter that the Cadzow-signal reconstruction combined with the new joint-horizon k-period-ahead forecasting algorithm did not always produce good forecasts. The reason for this is due to the "end-effects" that is caused by the Hankelization operation during the Cadzow-signal reconstruction.

It is also recommended that a simulation study be used to compare the results of k-period-ahead forecasts with the recurrent one-period-ahead methods and that of the joint-horizon method. Such a study would have to identify a few time series models of interest and use different noise models. Different lead-times for forecasts (k) should also be investigated as part of such a study.

Chapter 4

MODEL SELECTION IN SSA

4.1 Introduction

This Chapter contains the main results of this study. The entire Chapter is devoted to model selection techniques that were investigated in the SSA context.

Section 4.2 contains a survey of the relevant literature and briefly introduces the need for model selection in SSA, when forecasting is the end-goal.

There is mainly a choice of two "parameters" that has to be made in the SSA context. The first choice encompasses the window length (τ). Recall that the window length is simply the Euclidean dimension ($\mathbb{R}^{\tau}$) into which the column vectors of the trajectory matrix are unfolded. Section 4.3 is devoted to this topic.

The choice of window length is not by far as important as selecting the "correct" number of eigenvectors to use. The number of eigenvectors represents structure in a time series. A number of techniques are considered in Section 4.4. Techniques elaborated upon include the use of a scree-diagram, cumulative percentage variation explained by eigenvectors, artificial phase space portraits, cross- and forward validation. Two new forward validation approaches are proposed to assist in model selection.

An algorithm, that can be used for the identification of a single additive time series outlier, is proposed in Section 4.5.

A small scale Monte Carlo simulation study in Section 4.6 is used to show the effectiveness of the main algorithm proposed by this study. Conclusions are reached and recommendations are made in Section 4.8.

4.2 Literature Survey

This study set as goal the development of a model selection technique, in the SSA paradigm, that would automate model selection. The end-use of a model, selected using the model selection technique, is to assist in producing adequate forecasts. There are many reasons behind automation of selection procedures. Borrowing ideas from a time series setting, the following self-explanatory statement due to Gómez and Maravall (in Peña *et al.*, 2001, p. 172) outlines two good reasons.

"*There are many reasons why one should try to automate as much as possible the ARIMA model identification stage, but they can be basically reduced to two. The first one is that one should eliminate as much as possible all mundane and mechanical chores, which can be performed by the computer... The second reason has to do with the objectivity of the identification stage...*"

It must be made clear, *ab initio*, that a model selection technique must have the end-use, of the procedure under consideration, in sight. This study set as goal the development of a model selection technique, in the SSA paradigm, which would assist in producing adequate forecasts. The majority of model identification procedures, up to date, concentrated on the use of scree diagrams and artificial phase space portraits (Vautard and Ghil, 1989; Ghil and Vautard, 1991; Vautard *et al.*, 1992; Plaut and Vautard, 1994; Elsner and Tsonis, 1996; Golyandina *et al.*, 2001). The end-use of the latter techniques was not forecasting, but analysis of time series structure. Golyandina *et al.* (2001) introduced a method which they called "grouping". This method can also be placed in the paradigm of analysis of time series structure, even though their work also advocates forecasting in the SSA paradigm.

An exhaustive citation index search in this study yielded a single article that identified the need for model selection, in the SSA paradigm, with forecasting as end-use (Zhang *et al.*, 1993). This article pre-dates the advances due to Buchstaber (1994), Danilov (1997), Venter (1998) and Golyandina *et al.* (2001). The iterative forecasting procedure proposed by Zhang *et al.* (1993) became outdated with the formalization of the recurrent one-period-ahead forecast due to Danilov (1997). The most recent advance, addressing model selection with forecasting as end-use, is due to Venter (1998).

This study therefore attempts to produce a sound model selection technique, by incorporating recent advances in SSA-forecasting due to Buchstaber (1994), Danilov (1997), Venter (1998) and Golyandina *et al.* (2001).

Statisticians use a principle termed *parsimony* when considering candidate models. The idea of parsimony can be traced back to ideas formalised by Box and Jenkins (1970, p. 17), where they state that

"*It is important, in practice, that we employ the* ***smallest possible*** *number of parameters for adequate representation.*"

An excellent overview of parsimony can be found in Burnham and Anderson (1998, pp. 23-26). According to them

"*Statisticians view the principle of parsimony as a bias versus variance tradeoff. In general, bias decreases and variance increases as the dimension of the model (K) increases ...Often, we may use the number of parameters in a model as a measure of the degree of structure inferred from the data... Parsimonious models achieve a proper tradeoff between bias and variance.*"

The latter paragraph fits like a glove when considering model selection in the SSA paradigm. The structure of a "model", in SSA, is obtained from the number of eigenvectors incorporated in the model. There are no parameters in SSA since it encompasses non-parametric modelling.

The two main concerns in SSA are the selection of window length and number of eigenvectors that explain structure in a model. Attention will be given to the choice of these "parameters" in ensuing sections.

4.3 Choosing the Window Length

An excellent account that deals with the choice of window length can be found in Golyandina *et al.* (2001, Section 1.6.2). That section gives a thorough treatment of the choice of window length, when the extraction of certain components of a time series is of importance.

The window length (τ) essentially represents the dimension of the Euclidean space $(\mathbb{R}^{\tau})$ into which the column vectors of a trajectory matrix are embedded. The process of embedding, essentially transforms a univariate time series into a multivariate framework. In Chapter 2 it was mentioned that the choice of the window length is bounded by

$$2 \leq \tau \leq \text{integer part of } (N+1)/2. \tag{4.1}$$

The reason for this is natural and due to the SVD of the trajectory matrix. The trajectory matrix is essentially a Hankel structured matrix. This implies that there is symmetry between left and right eigenvectors, which are extracted from this type of structured matrix. Additional to the latter argument, is the fact that the rank of the trajectory matrix is bounded by the upper bound featuring in (4.1).

When considering the choice of window length in the SSA context, it is invaluable to note the following statement due to Ghil and Vautard (1991)

"*The window width ... represents a compromise between significant information and statistical confidence.*"

The latter statement implies that, in the absence of suitable knowledge of structure prevalent in a time series, a large window length is preferable. If more knowledge regarding prevailing structure in a time series is available, the choice of window length can be adjusted accordingly. One example of such knowledge regards inherent periodicities in a time series.

If there is any form of periodicity in a time series, careful consideration has to be given to the choice of window length. According to Vautard *et al*. (1992)

> *"SSA does not resolve periods longer than the window length."*

This is an extremely important observation, i.e. that the choice of window length is dictated by the periodicity present in an observed time series. Time series originating, for example, from climatologic, geophysical and astrophysical applications tend to have low frequency oscillations. It is not uncommon to find, for example, low frequency oscillations in the order of 20 years in global surface temperatures.

Golyandina *et al*. (2001, p. 74) recommend that the window length be set equal to a multiple of the periodicity present in a time series. This can, alternatively, be stated that the ratio of window length over periodicity be equal to an integer. In the case of a 20-year cyclical pattern, present in a time series of yearly data, a window length of $\tau \in \{20,\ 40,\ \ldots\} \leq \text{integer part of } \ (N+1)/2$ is therefore recommended.

If a time series exhibits a monthly seasonal variation, the choice of window length must conform to

$$\tau \in \{12,\ 24,\ \ldots\} \leq \text{integer part of } (N+1)/2\,. \tag{4.2}$$

It is, therefore, important to note that periodicity dictates a fixed window length. Other forms of structure, on the other hand, might dictate a large window length. If a combination of periodicity and trend is of importance, then both the latter arguments must be taken into consideration.

If SSA is considered from the perspective of dynamical systems analysis, then other procedures are used to select the window length. The choice of the window length is commonly addressed under the heading of choice of a proper time delay (Takens, 1981; Liebert and Schuster, 1989). Specialists in this field use an approach based on the so-called correlation integral. A brief summary of this approach can also be found in Section 1.2.1 in Golyandina *et al.* (2001).

Golyandina *et al.* (2001) state that the separability of functions is affected by the choice of window length. A counter example was supplied in Chapter 3 to illustrate how the Cadzow-signal can be used to overcome this phenomenon.

It is clear that the choice of window length is less important than selecting the "correct" number of eigenvectors in a SSA "model". The window length simply represents the Euclidean dimension into which the trajectory matrix column vectors are embedded. The choice of the number of eigenvectors is more crucial, since it is the number of eigenvectors that determine structure in the SSA "model". This issue is thoroughly considered in the following sections.

4.4 Choosing the Number of Eigenvectors

The previous section outlined the choice of a suitable window length. Aspects that deal with the choice of a number of eigenvectors are more structural and therefore of greater importance in SSA. This is especially the case when the end-use is forecasting.

4.4.1 Scree-diagrams

The use of a scree-diagram is a well-known dimension reduction technique in principal component analysis (Jolliffe, 1986; Jackson, 1991; Krzanowski, 2000). A definition of the scree-diagram is supplied in Definition 4.1 below.

Definition 4.1 Scree-diagram

A scree-diagram can be defined as a plot of ordered eigenvalues $\left(\lambda_{(i)}\right)$ against the reversed indices $(i = 1,\ldots,\tau)$. Hence a scree-diagram is a plot of the coordinate-pairs,

$\left\{\left(i,\ \lambda_{(\tau-i+1)}\right) \middle| i = 1,\ldots,\tau\right\}$.

A scree-diagram is traditionally used to select the number of principal components (PCs). During this process an elbow is identified in the scree-diagram. The number of eigenvectors used, then equals the index value at which the elbow occurred.

PCA is usually performed under a white noise error structure assumption. It was argued in Chapter 2 that the "noise-floor" in a scree diagram, resulting from white noise, is fairly flat. This is the main reason why an elbow is searched for in the scree-diagram. This reasoning can also be found in Golyandina *et al.* (2001, p. 66) where they state that

"*Another useful insight is provided by checking breaks in the eigenvalue spectra. As a rule, a pure noise series produces a slowly decreasing sequence of singular values. If such a noise is added to a signal, described by a few eigentriples with large singular values, then a break in the eigenvalue spectrum can distinguish signal eigentriples from the noise ones.*"

It is clear that the search for an elbow in the scree-diagram is described in the previous paragraph. This notion is also found in Krzanowski (2000, p. 68)

"*If one argues that those dimensions corresponding to the flat portion of this graph represent undifferentiable 'noise' components of the system, then one should logically choose r as being at the foot of the initial steep decline. This leads to the popular held belief that r should be chosen at the value i at which the 'elbow' of the scree diagram occurs...*"

It is clear that a scree-diagram therefore results in the use of a certain number, say r, of <u>leading eigenvalues</u>. This can be a dangerous choice in a heavily noise-distorted signal series. In such a case the noise "structure" might have an eigenvalue present amongst the top few ordered eigenvalues. Selection of the eigenvectors corresponding to the leading r eigenvalues might then possibly result in the inclusion of an eigenvector, which is due to noise "structure". This situation must be guarded against and therefore evaluation of the scree-diagram alone, is not an advisable practice in SSA and is, at best, a very subjective issue.

Example 4.1 Scree-diagram of the Airline Passenger Time Series

The airline passenger time series is considered in this example. The series exhibits both trend and seasonal variation. The series is of length $N = 144$. The choice of window length is therefore limited to

$$\tau \in \{12,\ 24,\ \ldots\} \leq \text{integer part of } (N+1)/2 = 72. \tag{4.3}$$

For purposes of this example the window length was set equal to $\tau = 72$. Centring of the trajectory matrix was used and SVD performed. Figure 4.1, below, exhibits the scree-diagram.

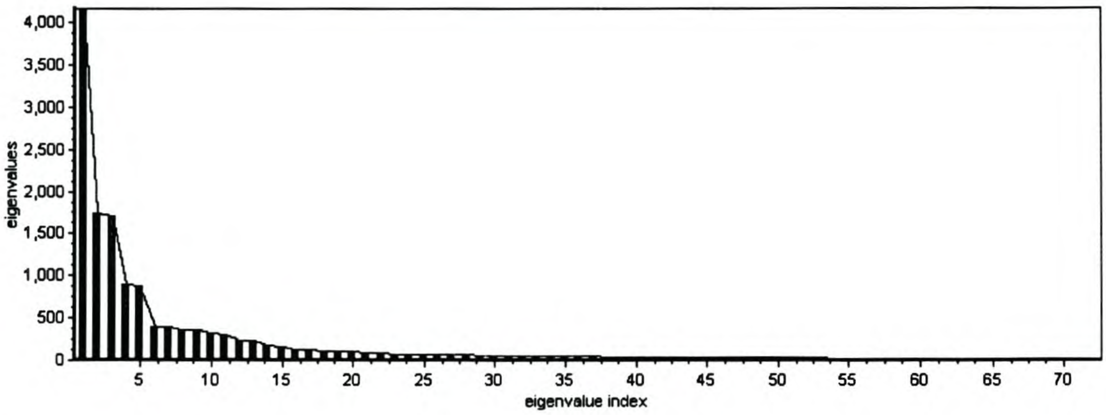


Figure 4.1 Scree-diagram of airline passenger time series. ($\tau = 72$ with centring used)

It would seem that an elbow is present at the index representing the 5 leading eigenvalues. Clearly, this elbow-choice is a fairly subjective one, and room for difference of opinion is possible.

Example 4.2 Scree-diagram of the Hotel Time Series

This second example considers the hotel series, which is of length $N = 168$. This series also exhibits trend combined with monthly seasonal variation. The choice of window length is therefore also bounded by

$$\tau \in \{12,\ 24,\ \ldots\} \leq \text{integer part of } (N+1)/2 = 84\,. \tag{4.4}$$

A window length of $\tau = 84$ was used and the resulting scree-diagram is exhibited in Figure 4.2 below.

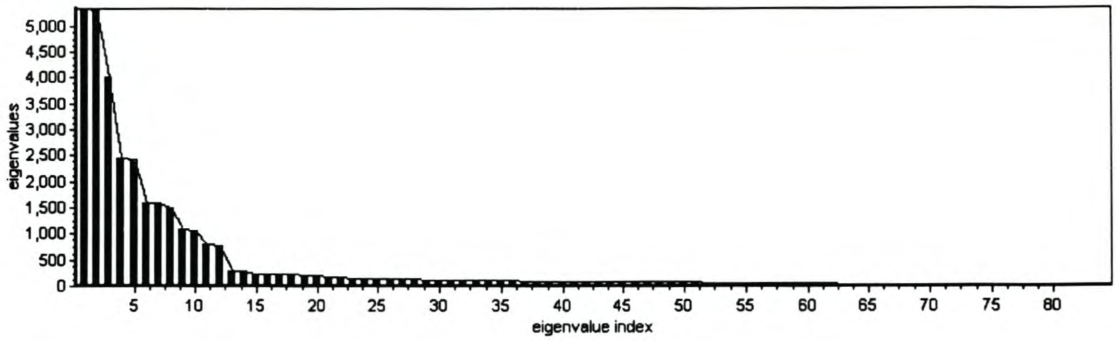


Figure 4.2 Scree-diagram of the hotel series. ($\tau = 84$ with centring used)

The 12 leading eigenvalues seem to be a more clear-cut choice in this case.

Example 4.3 Scree-diagram of the Liquor Time Series

The final example uses the liquor series, which is of length $N = 336$. This series contains some sort of polynomial trend and also monthly seasonal variation, bounding the choice of window length similarly to the previous examples. The window length was set at $\tau = 60$, not to over crowd the scree-diagram.

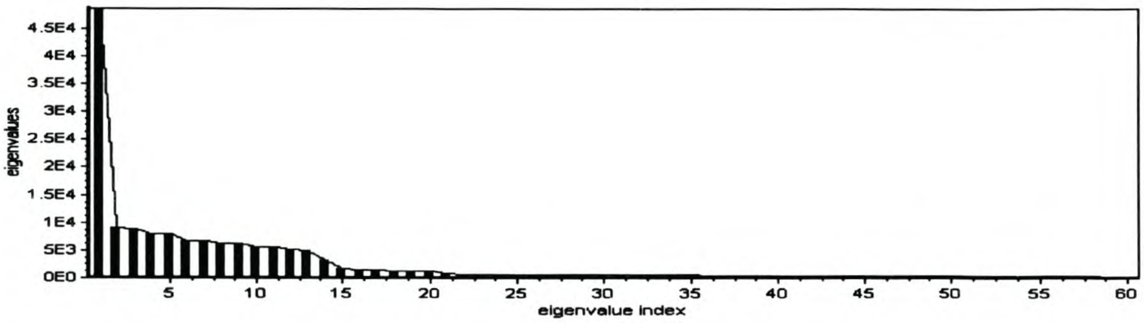


Figure 4.3 Scree-diagram of the liquor series. ($\tau = 60$ with centring used)

The leading eigenvalue seems to dominate in this series, judging by the eigenvalue spectra. The choice of a number of eigenvectors is also not easy. A subjective choice of 13 leading eigenvalues seems plausible.

4.4.2 Cumulative Percentage Variation Explained

Another popular method for selecting a number of PCs is the use of cumulative percentage variation explained by leading eigenvectors (Jolliffe, 1986; Jackson, 1991; Krzanowski, 2000). It is well known fact that the total variation of the scatter matrix is given by the trace of this matrix, and that it equals the sum of the eigenvalues, i.e. that

$$trace(\tilde{\mathbf{X}}\tilde{\mathbf{X}}') = \sum_{i=1}^{\tau} \lambda_{(i)} . \tag{4.5}$$

Some authors propose that the r PCs be used that have cumulative percentage variation explained in the range

$$80\% \leq \frac{\sum_{i=1}^{r} \lambda_{(\tau-i+1)}}{\sum_{i=1}^{\tau} \lambda_{(i)}} \times 100\% \leq 90\% . \tag{4.6}$$

Eigenvectors are used in SSA and not PCs. Hence, (4.6) can be used to select the r leading eigenvectors that correspond to the leading eigenvalues $\left(\lambda_{(i)}\right)$ with cumulative percentage variation as defined in (4.6). This study has, however, found this method ineffective due to the subjectivity involved.

There is one possible scenario where the above method can fail, which was also prevalent in the use of scree-diagrams. In the case of a heavily noise-dominated signal series, the noise "structure" might cause a large eigenvalue to feature amongst the ordered eigenvalues. Selecting a number of leading eigenvectors, based on cumulative percentage variation explained, will then also succumb to the selection of an eigenvector, which is due to noise "structure", rather than due to structure.

One possible solution of guarding against the above problem is using artificial phase space portraits when selecting eigenvectors.

4.4.3 Artificial Phase Space Portraits and Grouping

The use of artificial phase space portraits seems to be a very popular method for selecting eigenvectors in SSA. It is also this graphical method that leads to the "grouping" method proposed by Golyandina *et al.* (2001). According to Golyandina *et al.* (2001) the artificial phase space portraits are used in identifying pairs of eigenvectors that describe oscillatory behaviour in a time series. Slowly varying artificial phase space portraits, on the other hand, are used to identify trend in a time series. A very thorough treatment of the "grouping" method can be found in Section 1.6.1 of Golyandina *et al.* (2001).

It was proved in Chapter 2 that the r-flat on which the trajectory matrix column vectors reside, based on a sinusoidal signal series, are spanned by the vectors

$$\begin{aligned} \mathbf{y}_1 &= \left[\cos(\varpi t) \quad \cos(\varpi t + \Delta t\varpi) \quad \cdots \quad \cos[\varpi t + (\tau - 1)\Delta t\varpi]\right]' \\ \mathbf{y}_2 &= \left[\sin(\varpi t) \quad \sin(\varpi t + \Delta t\varpi) \quad \cdots \quad \sin[\varpi t + (\tau - 1)\Delta t\varpi]\right]', \end{aligned} \tag{4.7}$$

where ϖ is the frequency in radians and Δt the sampling interval. The artificial phase space portrait is a plot of the eigenvector pairing given by

$$\left\{\left(v_{1,j},\ v_{2,j}\right) \middle| j = 1,\dots,\tau\right\}. \tag{4.8}$$

In the case of a sinusoidal signal, the artificial phase space portrait will result in one of the figures exhibited in Figure 4.4 below.

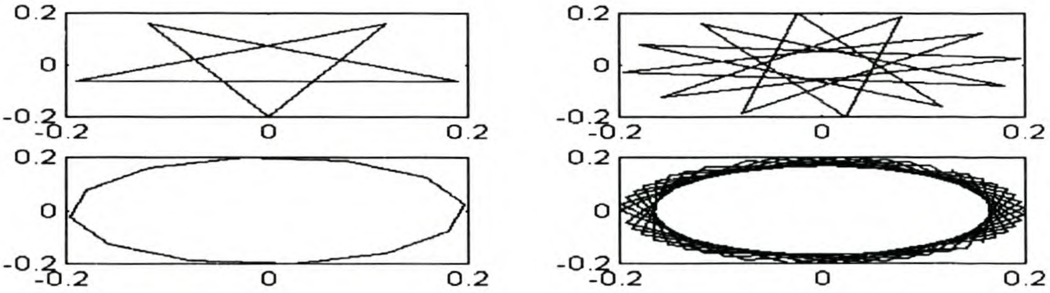


Figure 4.4 Artificial phase space portraits of sinusoids with different frequencies.

Another interesting fact that arises from oscillatory patterns is that $\lambda_{(k)} \approx \lambda_{(k+1)}$, i.e. that consecutive eigenvalues are fairly similar (Ghil and Vautard, 1991). This is another way of identifying oscillatory behaviour that can be identified through use of a scree-diagram or artificial phase space portraits (percentage variation explained by consecutive eigenvalues).

Example 4.4 Artificial Phase Space Portraits of the Airline Passenger Time Series

This example illustrates the use of "grouping" eigenvectors in the case of the airline passenger time series. The artificial phase space portrait of the airline passenger series is exhibited in Figure 4.5 below. A window length of $\tau = 72$ was chosen, under the assumption that no cyclical pattern with period higher that 72 is present in the series. Recall that SSA cannot resolve cyclical patterns with periods higher than the window length (cf. Section 4.3). Construction of the centred trajectory matrix, scatter matrix and SVD of the scatter matrix were performed. The resulting eigenvectors were used in the construction of the artificial phase space portraits.

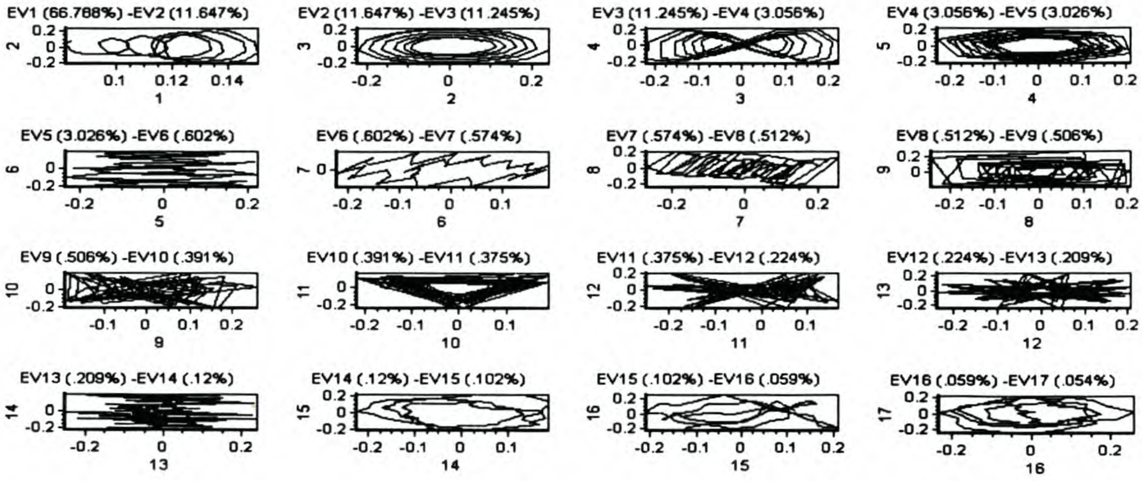


Figure 4.5 Paired phase space portraits of the airline time series.

From perusal of the above plot, the eigenvector pairings {(2,3), (4,5), (8,9), (10,11), (12,13)} seem plausible. All these parings have eigenvalues that have fairly similar percentage variation explained. The time series does not seem to be severely noise distorted, hence the 13 leading eigenvectors would possibly be a good choice. These leading eigenvectors have cumulative percentage variation explained equal to 99.305%.

The eigenvectors {1, 6, 7} explain trend in the series. To justify this observation, phase space portraits of non-paired eigenvectors must be considered. Figure 4.6, below, can be perused in this regard.

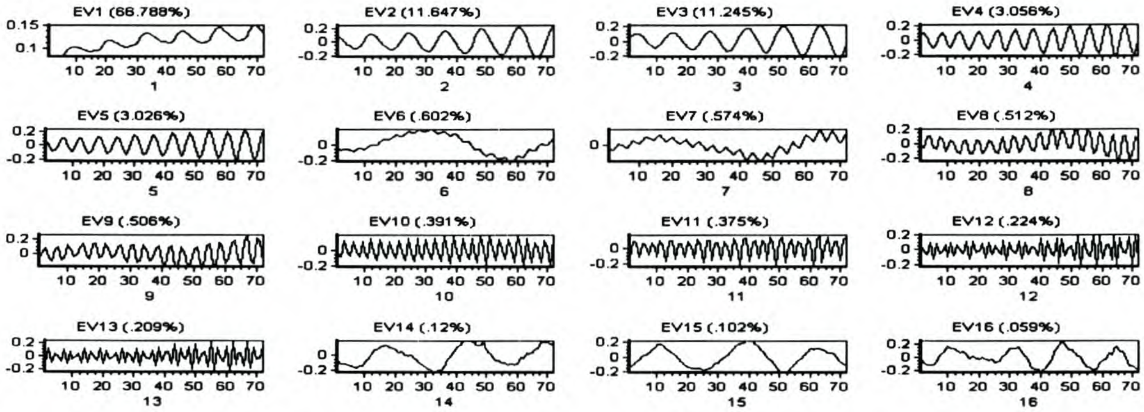


Figure 4.6 Individual phase space portraits of the airline time series.

Slowly varying eigenvectors explain trend. According to Golyandina *et al.* (2001, p. 56),

"*Thus, to extract a trend of a series, we have to collect all the elementary matrices related to slowly varying singular vectors.*"

In Figure 4.6, above, eigenvectors {1, 6, 7} conform to the description of slowly varying eigenvectors. These eigenvectors therefore contain the trend of the time series. The following "grouping" of eigenvectors is, according to the recommendations by Golyandina *et al.* (2001), plausible

$$\left\{ \overbrace{1}^{trend},\ \underbrace{\overbrace{(2,3)}^{cycle}}_{period=12},\ \underbrace{\overbrace{(4,5)}^{cycle}}_{period=6},\ \overbrace{6}^{trend},\ \overbrace{7}^{trend},\ \underbrace{\overbrace{(8,9)}^{cycle}}_{period=4},\ \underbrace{\overbrace{(10,11)}^{cycle}}_{period=3},\ \underbrace{\overbrace{(12,13)}^{cycle}}_{period=?} \right\}. \tag{4.9}$$

The periods of the sinusoids that correspond to the first 4 paired groupings can easily be deduced from the number of vertices in the polygons that each of these phase portraits represent. Note that this observation was already made in Chapter 2 (cf. Section 2.7.5).

It is not straightforward to deduce the period of the eigenvector pairing {12, 13}. A periodogram can be used here.

The airline series was next split into two parts, viz. a training series $\{x_t\}_{t=1}^{132}$ and test series $\{x_t\}_{t=133}^{144}$. A window length of $\tau = 60$ was used. This choice of window length is an integer multiple of the monthly seasonal period length of 12. Recurrent one-period-ahead forecasts and 12-period-ahead forecasts were generated over the test series. The results are summarised in Tables 4.1 to 4.4 below.

Table 4.1 Recurrent one-period-ahead forecasting results when no transformation, window length $\tau = 60$***, centring of the trajectory matrix and*** 13 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	31.34393	6.47%	1386.88566
Approximate series	33.50970	6.88%	1565.02387
Cadzow-signal	35.25708	7.16%	1734.02227

Table 4.2 Recurrent one-period-ahead forecasting results when a log-transformation, window length $\tau = 60$***, centring of the trajectory matrix and*** 13 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	29.46899	5.94%	1267.05261
Approximate series	32.09968	6.46%	1484.38595
Cadzow-signal	31.61045	6.40%	1405.70788

Table 4.3 12 ***-period-ahead forecasting results when no transformation, window length*** $\tau = 60$***, centring of the trajectory matrix and*** 13 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	34.63364	7.10%	1713.51262
Approximate series	37.72773	7.70%	1980.53734
Cadzow-signal	35.27855	7.17%	1736.20873
Vector forecasting	36.02855	7.43%	1738.49206

Table 4.4 12 ***-period-ahead forecasting results when a log-transformation, window length*** $\tau = 60$***, centring of the trajectory matrix and*** 13 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	30.84570	6.22%	1371.10356
Approximate series	34.70155	6.97%	1699.63154
Cadzow-signal	31.63199	6.40%	1407.22064
Vector forecasting	33.89824	6.85%	1547.39169

4.4.4 Cross Validation

Treatments that deal with Cross Validation (CV) as a model selection technique in a time series setting, can be found in McQuarrie and Tsai (1999) and Diebold (1998). The treatment is more thorough in the first reference. CV is a well researched variable selection method in the multiple regression context (Burnham and Anderson, 1998; Davison and Hinkley, 1999; Hjorth, 1999) and is also applied during smoothing and density estimation (Venables and Ripley, 1999; Thisted, 2000, p. 357)

Consider the class of autoregressive models

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + Z_t, \tag{4.10}$$

where

$$Z_t \overset{i.i.d.}{\sim} N\left(0,\ \sigma^2\right). \tag{4.11}$$

It is a well known fact that conditional least squares parameter estimates for the model in (4.10) can be found using multiple regression analysis. The conditional least squares parameter estimates are given by

$$\underset{(p\times 1)}{\hat{\boldsymbol{\phi}}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y}. \tag{4.12}$$

CV is a well-researched topic in the field of multiple regression analysis. Once an AR(p)-model is placed in this framework, results pertaining to model selection in a multiple regression setting can be invoked. This implies that CV can be used to select the order of the AR(p)-model, using existing theory derived in a multiple regression setting.

A natural question that arises when considering model selection techniques is, how can CV be adapted and applied in the SSA context? This section will do just that, adapt and apply CV in the SSA context.

CV leaves out a single time series observation, x_l, at a time and re-estimates it using the remaining time series observations. This principle is termed the leave-one-out (LOO) principle (Efron and Tibshirani, 1998, p. 240). If observation x_l is omitted in the SSA context, then it influences a number of reverse diagonals in the trajectory matrix. This is due to the Hankel structure of this matrix. The LOO principle will result in a single missing observation in a number of consecutive column vectors, in the trajectory matrix. These column vectors must be ignored when the shift vector $(\overline{\mathbf{x}})$ and scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}'\right)$ are created. The column vectors that contain a missing observation, due to the LOO principle, lie between and include columns j_1 to j_2, where

$$\begin{aligned} j_1 &= \max(1,\ l-\tau+1) \\ j_2 &= \min(l,\ n) \end{aligned} \tag{4.13}$$

and

$$\begin{aligned} l &= \text{index of the omitted time series observation}, 1 \leq l \leq N \\ \tau &= \text{window length} \\ n &= \text{total number of column vectors in trajectory matrix } \mathbf{X}. \end{aligned}$$

The affected block of column vectors is illustrated in (4.14) below. The column vectors appear between the dashed lines

$$\underset{(\tau \times n)}{\mathbf{X}} = \left[\begin{array}{cc|ccc|cc} x_{1,1} & \cdots & x_{1,j_1} & \cdots & x_{1,j_2} & \cdots & x_{1,n} \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{\tau,1} & \cdots & x_{\tau,j_1} & \cdots & x_{\tau,j_2} & \cdots & x_{\tau,n} \end{array}\right]. \tag{4.14}$$

All the unaffected column vectors in (4.14) are used to construct the shift vector and scatter matrix. The omitted observation is then estimated using a revised version of the one-period-ahead formula. This results in new column vectors $\hat{\mathbf{x}}_{j_1},\ldots,\hat{\mathbf{x}}_{j_2}$, all of which are residing on the same r-flat. These column vectors are replaced in the trajectory matrix in (4.14). The Hankelization operation is applied to the resulting matrix, resulting in a single time series. This time series is then smoothed using the Cadzow iterative property mapping algorithm. The observation $\hat{x}_l$ in the latter Cadzow-signal then constitutes the "prediction" of the left out observation x_l. The squared prediction error is then calculated and incorporated in a loss function, which will be used to select the best candidate model.

Algorithm 4.1, below, contains the necessary steps to follow when using CV in the context of SSA.

Algorithm 4.1 Cross Validation Algorithm in the SSA context

(a) Load an observed time series $\{x_t\}_{t=1}^{N}$ and transform if necessary.

(b) Loop over the window length, i.e. over $\tau_1 \leq \tau \leq \tau_2$. (Note that both τ_1 and τ_2 must be greater than the largest period length in the time series and are chosen arbitrarily. It is recommended that the choice of τ_2 be as large a possible, i.e. $\tau_2 \leq \text{integer part of } (N+1)/2$).)

(c) Construct the $\tau \times n$ trajectory matrix $\mathbf{X}$.

(d) Loop over the number of leading eigenvectors to use, i.e. over $1 \leq r \leq r_1$.

(e) Loop over the observation x_l to leave out, i.e. over $1 \leq l \leq N$.

(f) Calculate the indices of the column vectors in the trajectory matrix, created in step (c), that contain the omitted observation. These will be column vectors $j_1 = \max(1,\ l-\tau+1)$ to $j_2 = \min(l,\ n)$.

(g) Delete the column vectors j_1 to j_2 in the trajectory matrix of step (c). Use the remaining column vectors to construct the shift vector $(\bar{\mathbf{x}})$, centred trajectory matrix $(\tilde{\mathbf{X}})$ and scatter matrix $(\tilde{\mathbf{X}}\tilde{\mathbf{X}}')$.

(h) Perform SVD of the scatter matrix $(\tilde{\mathbf{X}}\tilde{\mathbf{X}}')$ and form the matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$, using the r leading eigenvectors based on the current loop in step (d).

(i) Construct an artificial l^{th} observation x_l in each of the column vectors where it was omitted using

$$x_l^{(j)} = \bar{x}_{l-j+1} + \frac{\left\langle \mathbf{P}_{\mathfrak{L}_r} \tilde{\mathbf{x}}_j^{(0,l-j+1)}, \mathbf{e}_{l-j+1} \right\rangle}{1 - \left\| \mathbf{P}_{\mathfrak{L}_r} \mathbf{e}_{l-j+1} \right\|^2}, \tag{4.15}$$

where

$$\begin{aligned} j_1 \leq j &\leq j_2 \\ \tilde{\mathbf{x}}_j^{(0,l-j+1)} &= j^{th} \text{ column vector of } \tilde{\mathbf{X}} \text{ with zero in position } (l-j+1) \\ \mathbf{e}_{l-j+1} &= \text{column vector } (l-j+1) \text{ of matrix } \mathbf{I}_\tau \ . \end{aligned}$$

(j) Average over the reverse diagonals of the resulting trajectory matrix that contains the reconstructed column vectors $\hat{\mathbf{x}}_{j_1},\dots,\hat{\mathbf{x}}_{j_2}$. This operation is essentially the Hankelization operation, and results in a single time series.

(k) Smooth the series in step (j) using the Cadzow-signal, using the window length and number of leading eigenvectors in the current loop (cf. Algorithm 2.2 in Chapter 2).

(l) Use as prediction of x_l the l^{th} element of the Cadzow-signal of step (k). Calculate the squared prediction error $\left(x_l - \hat{x}_l\right)^2$.

(m) When the inner loop in step (e) has calculated squared prediction errors for all the time series observations, calculate the mean square error and store this in a matrix $\boldsymbol{MSE}$ as follows,

$$\underset{(\tau_2-\tau_1+1)\times r_1}{\mathbf{MSE}} = \begin{bmatrix} mse_{\tau_1,1} & mse_{\tau_1,2} & \cdots & mse_{\tau_1,r_1} \\ \vdots & \vdots & \vdots & \vdots \\ mse_{\tau_2,1} & mse_{\tau_2,2} & \cdots & mse_{\tau_2,r_1} \end{bmatrix} \tag{4.16}$$

where

$$mse_{i,j} = \frac{1}{N}\sum_{l=1}^{N}\left(x_l - \hat{x}_l\right)^2 \tag{4.17}$$

for $i = \tau_1,\dots,\tau_2$ and $j = 1,\dots,r_1$.

(n) The program continues until the loop in steps (d) and (b) have been completed. The **MSE** matrix in (4.16) will then be filled with mean squared error values. The following bias-variance tradeoff measure is then calculated,

$$CV_j = \underbrace{\left[\frac{1}{\tau_2 - \tau_1 + 1}\sum_{i=\tau_1}^{\tau_2} mse_{i,j}\right]^2}_{\text{mean(MSE)}^2} + \underbrace{\frac{1}{\tau_2 - \tau_1}\sum_{i=\tau_1}^{\tau_2}\left[mse_{i,j} - \left\{\frac{1}{\tau_2 - \tau_1 + 1}\sum_{i=\tau_1}^{\tau_2} mse_{i,j}\right\}\right]^2}_{\text{var(MSE)}} \tag{4.18}$$

for $j = 1,\ldots,r_1$.

The above CV approach is due to Venter (1998), but has been adapted in this study to include a bias-variance tradeoff. Steps (m) and (n) have been introduced for this purpose. The essentiality of a bias-variance tradeoff, during model selection, was elaborated upon in Section 4.2. The bias variance tradeoff is prevalent in the above algorithm in the form of (4.18).

The end-goal of model selection in the SSA paradigm is to select a "model", which translates into selecting a number of leading eigenvectors $(j \in \{1,\ldots,r_1\})$ in Algorithm 4.1. Recall that eigenvectors represent structure in the SSA context. In order to create a "sample" of MSE values for each "candidate model", Algorithm 4.1 loops over the window length in step (b). This creates an artificial "sample" of MSE values $\{mse_{i,j} \,|\, i = \tau_1,\ldots,\tau_2\}$ for each "candidate model", i.e. $j \in \{1,\ldots,r_1\}$. Two things are of importance when considering the MSE values during model selection. These values should preferably be close to zero and not vary too much over the choice of window length $(\tau \in \{\tau_1,\ldots,\tau_2\})$. Hence the mean squared error of the MSE is used in this regard, cf. (4.18). This incorporates both bias and variance during the model selection.

Two examples are considered next for illustrative purposes, using the CV method described above. This study did not find CV to be an effective model selection technique. The use of the Cadzow-signal, during the CV method, is extremely expensive on CPU time. Another drawback with the CV approach is that the observations are all given the same weight in this scheme. It should be clear that recent observations should carry more weight, when using a model selection method with forecasting as end-goal. The models that are selected, using the number of leading eigenvectors, also do not honour the pairings that might be natural to the time series. This issue will be addressed at a later stage.

Example 4.5 Cross validation model selection in the case of the airline passenger time series

The airline passenger time series was used for the purposes of this example. Recall that the series is of length $N = 144$. The time series was log-transformed to deal with heteroskedasticity. The series was split into two parts, i.e. a training set $\{x_t\}_{t=1}^{132}$ and test set $\{x_t\}_{t=133}^{144}$. The test set was used to test the selected model's out-of-sample forecasting accuracy.

Cross validation was performed over the training series, using Algorithm 4.1. The window length loop was set to loop over $15 \leq \tau \leq 66$. The loop over the number of leading eigenvectors was set to loop over $1 \leq r \leq 13$. The CV analysis used 5 hours and 34 minutes of CPU time on a personal computer fitted with an 800mHz processor. The CV results are summarised in Table 4.5 below.

Figure 4.7, below, depicts the CV results. Forecasting over the test series, using the 13 leading eigenvectors, has already been dealt with in Example 4.4, and can be consulted for the out-of-sample forecasting accuracy of this choice of model.

Table 4.5 Summary of cross validation model selection results for the airline passenger time series.

Number of leading eigenvectors (r)	CV_r (bias^2 + variance)	Mean(MSE) (bias)	Var(MSE) (variance)
13	6167.132	74.560	607.956
12	13409.769	113.991	415.855
11	14107.840	117.171	378.813
9	29612.065	171.552	181.816
10	30640.011	174.188	298.470
7	37796.715	193.561	330.841
8	58706.356	241.783	247.276
5	58738.432	242.128	112.460
6	59518.340	243.835	62.863
4	284992.773	533.772	80.733
3	308177.051	555.118	20.871
2	2948419.885	1717.089	26.455
1	3056494.030	1748.282	5.802

Table 4.5, above, summarises the cross validation results. The first column consists of the number of leading eigenvectors used in each "model". These results are ordered in an ascending fashion, according to the measure CV_r which was proposed in (4.18). The bias and variance then follows and were calculated using

$$\begin{aligned}\text{bias}_\text{j} &= \frac{1}{\tau_2-\tau_1+1}\sum_{i=\tau_1}^{\tau_2} mse_{i,j} \\ &= \frac{1}{52}\sum_{i=15}^{66} mse_{i,j}\end{aligned} \tag{4.19}$$

and

$$\begin{aligned}\text{variance}_\text{j} &= \frac{1}{\tau_2-\tau_1}\left[\sum_{i=\tau_1}^{\tau_2} mse_{i,j} - \left\{\frac{1}{\tau_2-\tau_1+1}\sum_{i=\tau_1}^{\tau_2} mse_{i,j}\right\}\right]^2 \\ &= \frac{1}{51}\left[\sum_{i=15}^{66} mse_{i,j} - \left\{\frac{1}{52}\sum_{i=15}^{66} mse_{i,j}\right\}\right]^2 .\end{aligned} \tag{4.20}$$

It is evident from perusal of Table 4.5, above, that bias decreased and that variance increased as more leading eigenvectors were used. The single leading eigenvector has the lowest variability, over the window length, but has the largest bias of all the models. This eigenvector describes the trend in the airline time series. As the number of leading eigenvectors was increased, the bias decreased rapidly and seems to tend towards a saturation point. Care and judgement should be used in such case, as not to over parameterise the model.

The best candidate model seems to be the one that contains 13 leading eigenvectors.

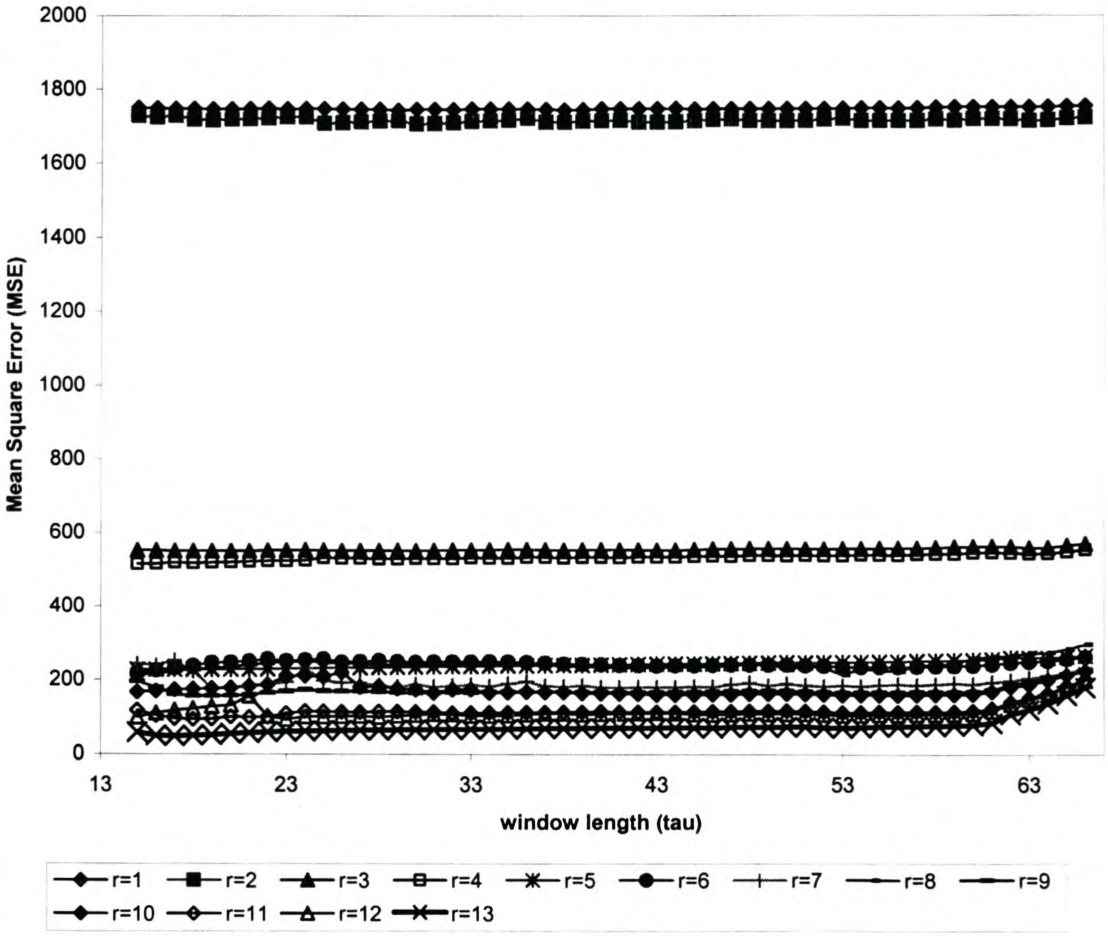


Figure 4.7 Cross validation results of airline passenger time series.

Example 4.6 Cross validation model selection for the hotel time series

The hotel time series was used for the purposes of this example. Recall that the series is of length $N = 168$. The series was split into two parts, i.e. a training set $\{x_t\}_{t=1}^{156}$ and test set $\{x_t\}_{t=157}^{168}$.

Cross validation was performed over the training series, using Algorithm 4.1. The window length loop was set to loop over $15 \leq \tau \leq 78$. The loop over the leading eigenvectors was set to loop over $1 \leq r \leq 13$. The CV analysis used 8 hours and 18 minutes of CPU time on a personal computer fitted with an 800mHz processor.

Table 4.6 Summary of cross validation model selection results of hotel time series.

Number of leading eigenvectors (r)	$\mathbf{CV_r}$ (bias^2 + variance)	Mean(MSE) (bias)	Var(MSE) (variance)
11	159896.315	397.564	1839.148
12	166987.987	289.622	83107.285
10	183853.078	428.316	398.665
13	508552.995	342.096	391523.449
9	589797.842	767.787	301.117
8	663591.407	814.485	205.748
7	1641839.276	1271.245	25775.218
6	3595748.773	1894.405	6978.488
5	3682572.794	1918.935	261.959
4	12561021.985	3544.073	569.741
3	13143121.115	3625.310	250.474
2	136493640.478	11650.679	755320.471
1	216034918.354	14293.344	11735236.109

A few interesting facts can be deduced from perusal of Table 4.6 above. As the number of leading eigenvectors in a model was increased, bias generally decreased and little pattern in the behaviour of the variance was noticed. This was also found in the case of the airline passenger time series. The best candidate model has 11 leading eigenvectors.

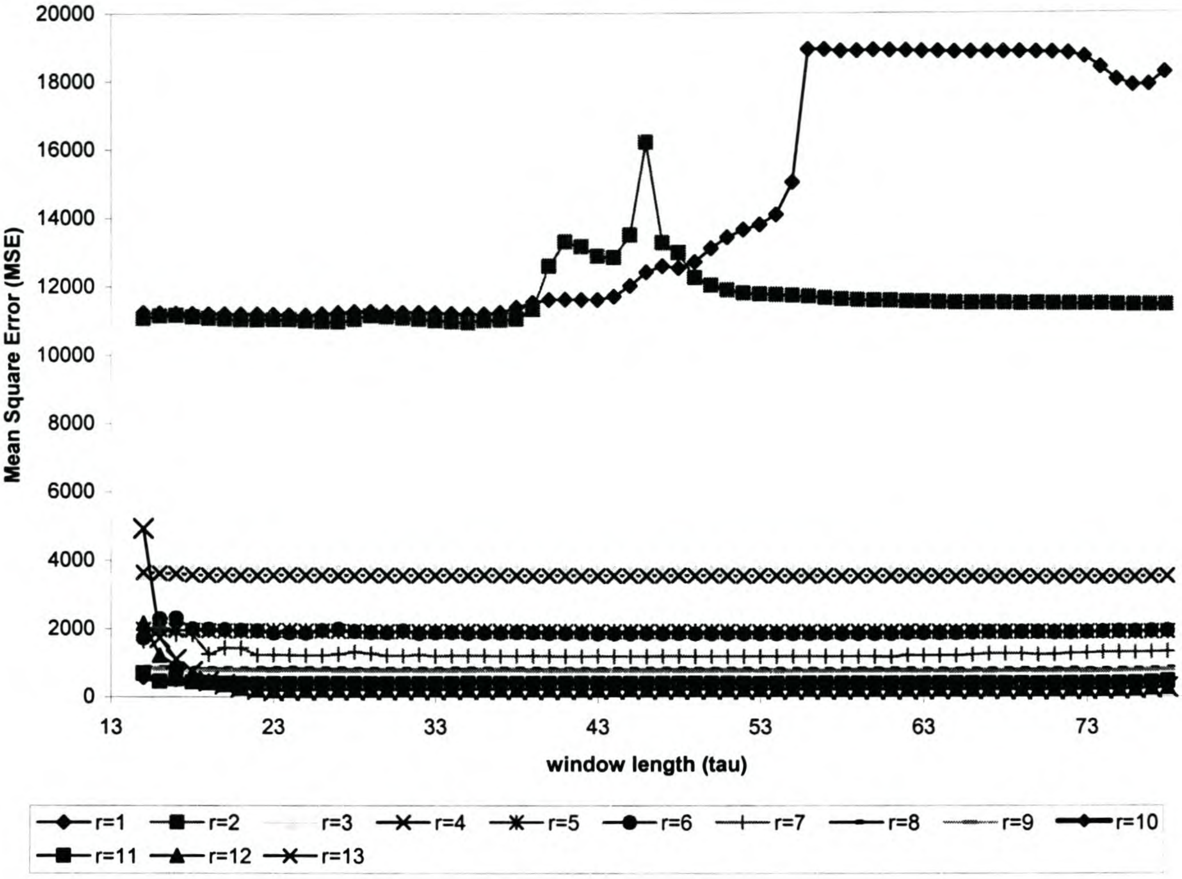


Figure 4.8 Cross validation results of the hotel time series.

4.4.5 Forward Validation

The eventual use of a candidate model influences the choice of model selection technique. If forecasting is the end-goal, then forward validation is a better model selection technique in the SSA paradigm. Forward validation captures the "structural dynamics" of a model better than cross validation. CV treats each of the observations, left out during the validation process, as equally important. In the context of forecasting it is *prima facie* that the most recent observations in a time series are usually deemed to be more important. The more recent the observation, the greater weight it should carry in construction of a forecast.

Forward validation falls in the broad class of computer intensive techniques. According to Thisted (2000, pp. 227-228)

"The term 'computer-intensive methods' was coined by Efron and Diaconis (1983) to refer to sample-reuse methods such as the bootstrap, jackknife and cross-validation... The term now encompasses a wide range of methods, which, roughly speaking, relinquishes standard assumptions of classical statistics such as normal error distributions or linearity of response at the cost of massively increased computational effort. Many of the statistical and computational ideas now associated with this term have been developed by Jerome Friedman and his colleagues and students at the Stanford Accelerator Centre."

The algorithms that are proposed in this section are heavily reliant upon CPU speed. More advanced problems can be handled as the capabilities of computer hardware increases in leaps and bounds. Gordon Moore, a co-founder of Intel Corporation which is responsible for the well-known Intel range of processors (Pentium), predicted that microprocessor speed would double every eighteen months. This statement was already made in 1965 and is today known as Moore's Law. At the time of writing this thesis, IBM introduced the world's fastest CPU running at 110 gigaHerz. CPUs that are available for personal computers are already clocking 2 gigaHerz in the form of Intel's Pentium 4 processor range.

Forward validation is, *inter alia*, used by the neural network (NN) community. In the NN framework, decisions must be made with regards to the number of layers to be used in the neural network architecture (Weigend *et al.*, 1990; Kuan and Liu, 1995). The number of layers is chosen based on the performance of the layers, which is indirectly measured through some loss function. SSA and NN have this in common, that these techniques are definitely non-parametric by nature.

FV splits an observed time series into three non-overlapping sets, termed the training-, validation- and test sets (Weigend *et al.*, 1990). The idea is exhibited in Figure 4.9 below.

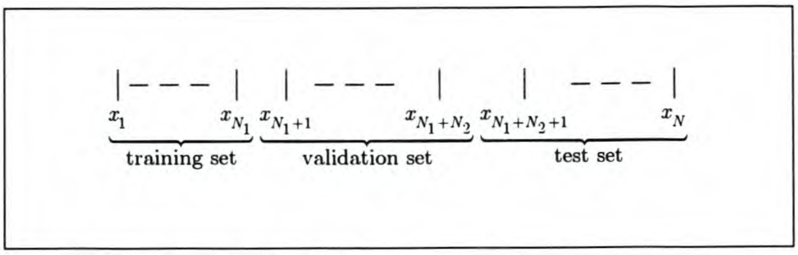


Figure 4.9 Forward validation time series.

Some authors (Diebold, 1998, pp. 117-118) state that FV splits an observed time series only into two sets, viz. the training set and validation set. This study uses the notion due to Weigend *et al.* (1990). It is a well-known fact that a fitted time series' in-sample and out-of-sample measures of forecasting accuracy differ. The test set, proposed by Weigend *et al.* (1990), will therefore be used to test a candidate model's out-of-sample performance.

We now propose a new FV method in Algorithm 4.2 below. This algorithm performs rolling joint-horizon k-period-ahead forecasting with re-estimation during FV model selection. It has been found that the algorithm produces stable results as k, the number of periods being forecast ahead, increases. Venter (1998) proposed FV in the SSA context using one-period-ahead forecasting and a different approach than the one presented in Algorithm 4.2. His approach will not be dealt with in this study.

Algorithm 4.2 Forward validation using rolling k-period-ahead forecasting with re-estimation

(a) Load an observed time series $\{x_t\}_{t=1}^{N}$ and transform if necessary.

(b) Set the number of periods to forecast ahead (k). Split the time series into a training set $\{x_t\}_{t=1}^{N_1-k+1}$, validation set $\{x_t\}_{t=N_1+1}^{N_1+N_2}$ and test set $\{x_t\}_{t=N_1+N_2+1}^{N}$.

(c) Set the maximum number of leading eigenvectors $r_{\max}$ and the maximum window length $\tau_{\max}$($\leq$ integer part of $(N_1-k+2)/2$). Loop over the window length (τ), where $k+r_{\max}\leq\tau\leq\tau_{\max}$.

(d) Loop over the number of leading eigenvectors to use, where $1\leq r\leq r_{\max}$.

(e) Construct the Cadzow-signal series using the training set, the window length in the loop at step (c) and the number of leading eigenvectors in the loop in step (d).

(f) Unfold the Cadzow-signal series into the column vectors of a $\tau \times (N_1 - k - \tau + 2)$ trajectory matrix. Construct the centred trajectory matrix $\tilde{\mathbf{X}}$ and scatter matrix $(\tilde{\mathbf{X}}\tilde{\mathbf{X}}')$.

(g) Perform SVD of the scatter matrix $(\tilde{\mathbf{X}}\tilde{\mathbf{X}}')$.

(h) Construct the matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$, using the r leading eigenvectors.

(i) Construct the projection matrix $\mathbf{P}_{\mathfrak{L}_r}$.

(j) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r^\perp} = \mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}$.

(k) Partition the projection matrix in (j) as follows

$$\begin{aligned} \mathbf{P}_{\mathfrak{L}_r^\perp} &= \left[\mathbf{P}_{\mathfrak{L}_r^\perp} \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-k} \end{bmatrix} \middle| \mathbf{P}_{\mathfrak{L}_r^\perp} \begin{bmatrix} \mathbf{e}_{\tau-k+1} & \cdots & \mathbf{e}_\tau \end{bmatrix} \right] \\ &= \left[\underset{\tau\times(\tau-k)}{\mathbf{P}_1} \middle| \underset{\tau\times k}{\mathbf{P}_2} \right], \end{aligned} \tag{4.21}$$

where $\mathbf{e}_j$ is the j^{th} column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and k is the number of periods being forecast ahead.

(l) Calculate the joint-horizon k-period-ahead forecasts using

$$\begin{bmatrix} \hat{x}_{N_1-k+2} \\ \vdots \\ \hat{x}_{N_1+1} \end{bmatrix} = \begin{bmatrix} \bar{x}_{\tau-k+1} \\ \vdots \\ \bar{x}_\tau \end{bmatrix} - \left(\mathbf{P}_2'\mathbf{P}_2\right)^{-1}\left(\mathbf{P}_2'\mathbf{P}_1\right) \begin{bmatrix} x_{N_1-\tau+2} - \bar{x}_1 \\ \vdots \\ x_{N_1-k+1} - \bar{x}_{\tau-k} \end{bmatrix}, \tag{4.22}$$

where

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{P}_{\mathfrak{L}_r^\perp} \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_{\tau-k} \end{bmatrix} \\ \mathbf{P}_2 &= \mathbf{P}_{\mathfrak{L}_r^\perp} \begin{bmatrix} \mathbf{e}_{\tau-k+1} & \cdots & \mathbf{e}_\tau \end{bmatrix}. \end{aligned} \tag{4.23}$$

(m) If a transformation was applied to the series in step (a) then a back-transformation must be used in this step. Calculate the mean squared prediction error

$\frac{1}{k}\sum_{i=1}^{k}\left(x_{N_1-k+i+1}-\hat{x}_{N_1-k+i+1}\right)^2$. Increase the training set length to $N_1 = N_1 + 1$ and return to step (e). The process is continued until the whole validation set has been forecast k-periods-ahead.

(n) Form elements of the following matrix

$$\underset{(\tau_{\max}-k-r_{\max}+1)\times r_{\max}}{\mathbf{MSE}} = \begin{bmatrix} mse_{k+r_{\max},1} & mse_{k+r_{\max},2} & \cdots & mse_{k+r_{\max},r_{\max}} \\ mse_{k+r_{\max}+1,1} & mse_{k+r_{\max}+1,2} & \cdots & mse_{k+r_{\max}+1,r_{\max}} \\ \vdots & \vdots & \vdots & \vdots \\ mse_{\tau_{\max},1} & mse_{\tau_{\max},2} & \cdots & mse_{\tau_{\max},r_{\max}} \end{bmatrix}, \quad (4.24)$$

where

$$mse_{i,j} = \frac{1}{kN_2}\sum_{t=N_1+1}^{N_1+N_2}\sum_{i=1}^{k}\left(x_{t-k+i}-\hat{x}_{t-k+i}\right)^2 \quad (4.25)$$

for $k + r_{\max} \le i \le \tau_{\max} (\le \text{integer part of } (N_1 - k + 2)/2)$ and $1 \le j \le r_{\max}$.

(o) Calculate the following measure

$$FV_j = \underbrace{\left[\frac{1}{\tau_{\max}-r_{\max}-k+1}\sum_{i=k+r_{\max}}^{\tau_{\max}} mse_{i,j}\right]^2}_{\text{mean(MSE)}^2}$$

$$+\underbrace{\frac{1}{\tau_{\max}-r_{\max}-k}\sum_{i=k+r_{\max}}^{\tau_{\max}}\left[mse_{i,j}-\left\{\frac{1}{\tau_{\max}-r_{\max}-k+1}\sum_{i=k+r_{\max}}^{\tau_{\max}} mse_{i,j}\right\}\right]^2}_{\text{var(MSE)}}$$

(4.26)

for $1 \le j \le r_{\max}$.

Remark 4.1

(a) One-period-ahead forecasting during FV is a special case of the above algorithm. One-period-ahead forecasting can be used during FV if a model is required that produces one-period-ahead forecasts. It is, however, suggested that $k > 6$ be used in step (b). Using a k-period-ahead forecast can be likened to fitting a discretely sampled piecewise smooth curve of length k to a section of an observed time series. During FV this is performed for the whole validation series of length N_2, i.e. N_2 forecasts of k-periods ahead are generated.

(b) The training set in step (b) of Algorithm 4.2 is used to form a Cadzow-signal series. This causes the algorithm to be expensive on CPU time. Two other alternatives exist here. The original training series can be used or the approximate series, which was introduced in Chapter 3. The approximate series is similar to the Cadzow-signal series. To reconstruct this series, only a single iteration in the scheme is used. Using the approximate series is not as expensive as the Cadzow-signal series on CPU time. The Cadzow-signal is, however, to be preferred as it ensures that the reconstructed signal series conforms to the two underlying assumptions of SSA. Another benefit of using a reconstructed Cadzow-signal series is that this series is robust against outliers that might possible be present in the training series.

(c) It is clear that the above algorithm uses re-estimation during the forward validation process. When a new time series observation is added to the training set, the whole SSA scheme is run through again to produce a k-period-ahead forecast. This feature is termed rolling forecasting with re-estimation, and is becoming increasingly popular in automated forecasting software (Ord and Lowe, 1996).

The effectiveness of the above algorithm will now be illustrated through a few examples.

Example 4.7 Forward validation model selection in case of the airline series using* 1*-period-ahead forecasting

Algorithm 4.2, above, was used in this example. A log-transformation was applied to the airline time series. The number of periods being forecast ahead was set at $k = 1$. The airline series was then split into three sets, i.e. training set $\{x_t\}_{t=1}^{120}$, validation set $\{x_t\}_{t=121}^{132}$ and test set $\{x_t\}_{t=133}^{144}$. The maximum number of leading eigenvectors used was set at $r_{\max} = 13$. The FV analysis used 20 minutes of CPU (800mHz) time. The FV results are summarized in Figure 4.10 and Table 4.7 below.

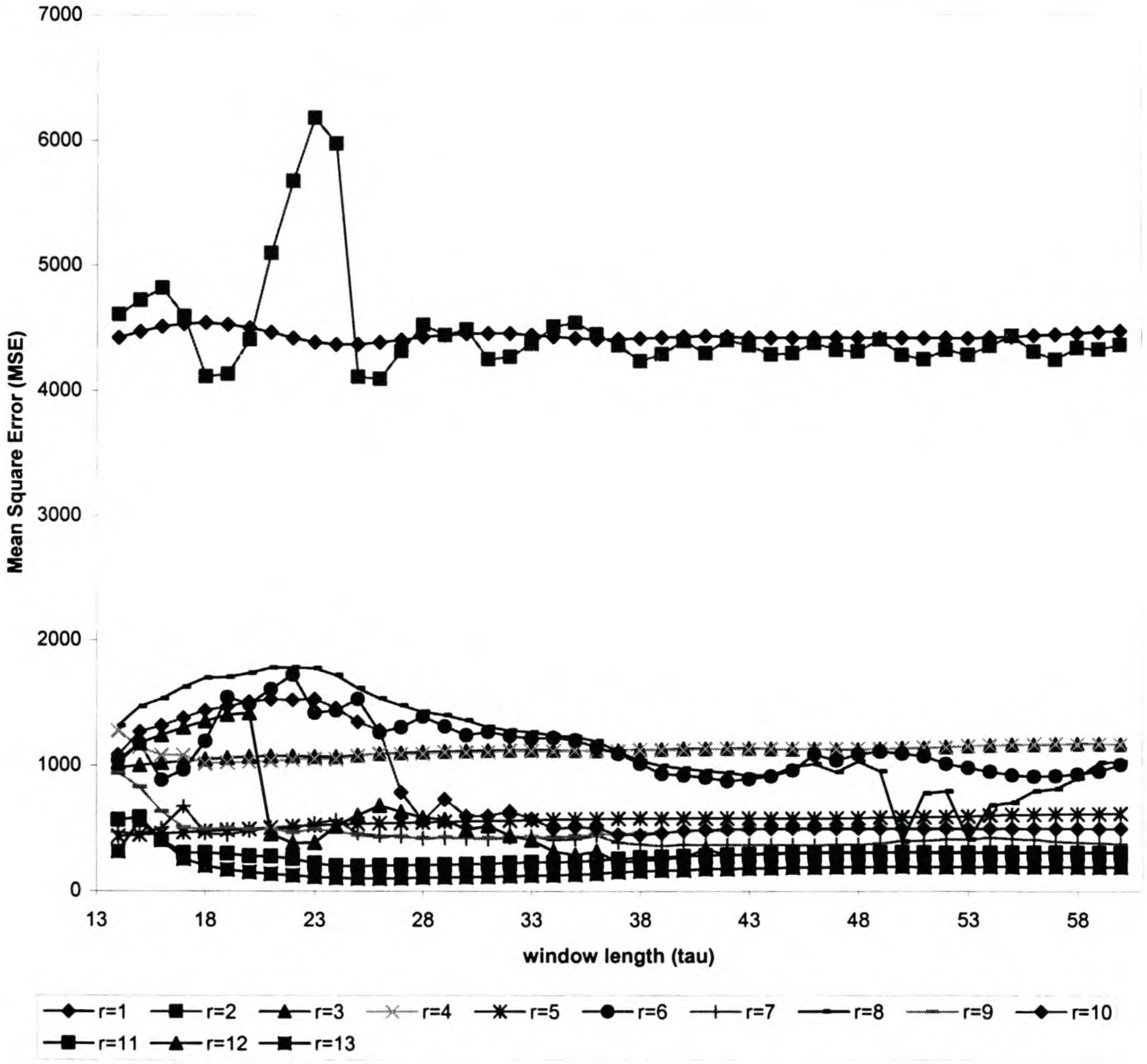


Figure 4.10 Forward validation results of the airline series.

Table 4.7 Forward validation results of the airline series when using* 1*-period-ahead forecasting.

Number of leading eigenvectors (r)	FV$_r$ (bias^2 + variance)	Mean(MSE) (bias)	Var(MSE) (variance)
13	38293.721	178.929	6278.114
11	88022.962	287.152	5566.954
7	184067.295	424.782	3627.391
9	252885.492	494.566	8290.196
5	313009.875	557.390	2325.738
12	374855.755	504.824	120008.498
10	749023.706	765.058	163710.450
3	1250342.108	1117.203	2199.339
4	1261416.271	1122.070	2375.192
6	1346344.615	1139.845	47098.722
8	1539815.934	1185.814	133659.997
1	19698060.392	4438.092	1404.106
2	20263855.550	4481.094	183655.599

The model that has the lowest FV measure, used 13 leading eigenvectors. This is coincidentally the same choice that CV yielded. The analysis used only 20 minutes of CPU (800mHz) time, which compares very favourable against the 5 hours and 34 minutes that the CV approach used. It must also be mentioned that the window length looped over, was $13 \le \tau \le 60$ in the case of this example.

The following example performed the same FV model selection, as the present example, but used 12 -period-ahead forecasting during FV.

Example 4.8 Forward validation model selection in the case of the airline series using 12 ***-period-ahead forecasting***

The airline time series was again used for the purposes of this example. The series was log-transformed. The number of periods being forecast ahead was set at $k = 12$. The airline series was split into three sets, i.e. training set $\{x_t\}_{t=1}^{109}$, validation set $\{x_t\}_{t=121}^{132}$ and test set $\{x_t\}_{t=133}^{144}$. The maximum number of leading eigenvectors used was set at $r_{\max} = 13$. The window length looped over during FV was $24 \leq \tau \leq 55$. The FV analysis used 8 minutes and 22 seconds of CPU (800mHz) time.

The FV results are summarised in Figure 4.11 and Table 4.8 below.

The analysis took less time than FV using 1-period-ahead forecasting. The reason is the fact that the window length, which was looped over, contained fewer values. The choice of the window length is dictated by the number of periods being forecast ahead. It is interesting to note that the model selection results do not seem to differ too substantially. The top two models are exactly the same and respectively used 13 and 11 leading eigenvectors.

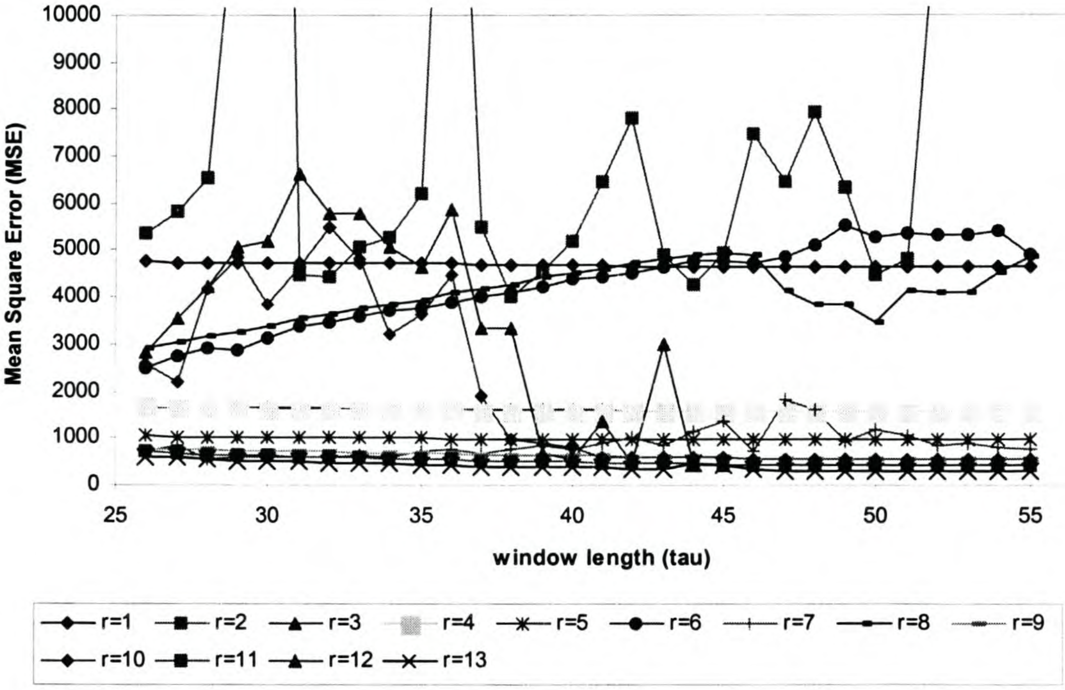


Figure 4.11 Forward validation results of the airline series.

Table 4.1 to Table 4.4 on page 175 can be consulted for the out-of-sample forecasting accuracy over the test set $\{x_t\}_{t=133}^{144}$, of a model using window length $\tau = 60$ and 13 leading eigenvectors.

Table 4.8 Forward validation results for the airline series when using* 12 *-period-ahead forecasting.

Number of leading eigenvectors (r)	FV_r (bias^2 + variance)	Mean(MSE) (bias)	Var(MSE) (variance)
13	177703.915	408.848	10547.230
11	274478.271	514.158	10119.815
9	407892.611	634.141	5757.214
7	845855.553	871.716	85967.108
5	971820.470	985.444	719.789
4	2512330.431	1584.425	1928.018
3	2546006.103	1595.227	1257.750
10	6659128.719	1923.274	2960146.587
12	10611023.660	2404.601	4828917.232
8	16540876.003	4020.000	380478.383
6	18357639.829	4170.098	967924.900
1	21957424.538	4685.717	1476.106
2	260410450839111.000	2860810.920	252226211721137.000

Figure 4.12, below, exhibits the combined training and validation sets in a single series $\{x_t\}_{t=1}^{132}$ and the reconstructed Cadzow-signal series. The Cazow-signal series was reconstructed using a window length of $\tau = 60$, and made use of the 13 leading eigenvectors.

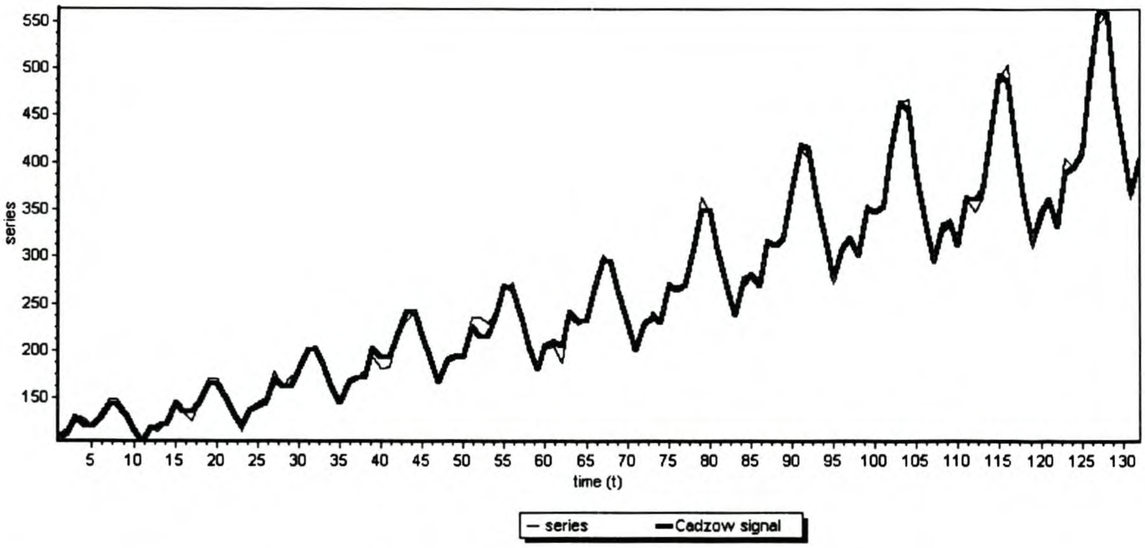


Figure 4.12 Airline series and reconstructed Cadzow-signal series. (*$\tau = 60$*,* $r = 13$*)

Figure 4.13, below, exhibits the residual series, which resulted from subtracting the reconstructed Cadzow-signal series from the airline series. It is clear that the residual series is stationary and does not seem to contain structure, in the form of periodicity or trend.

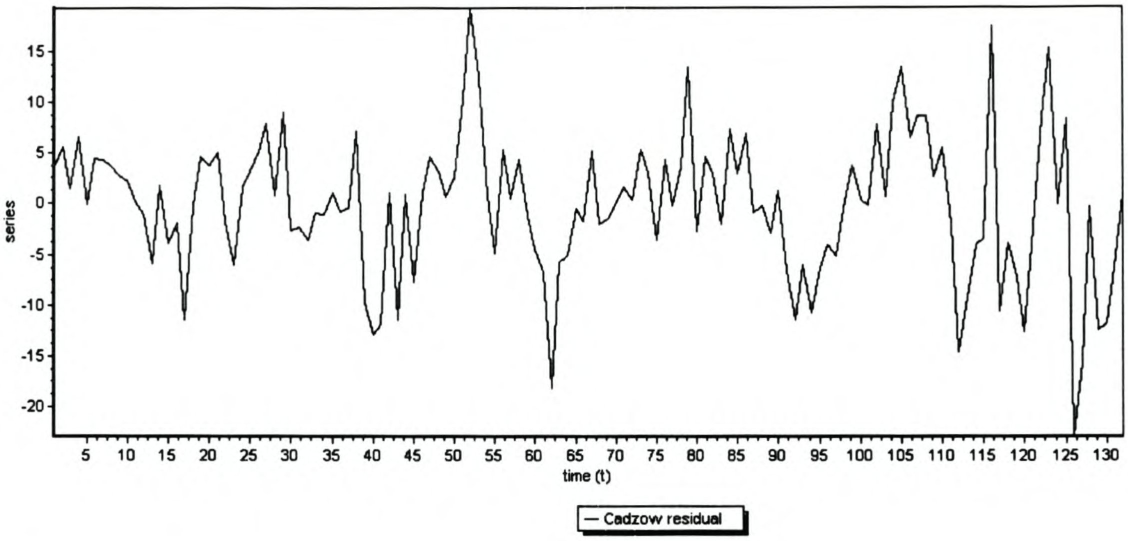


Figure 4.13 Residual series of the airline series and Cadzow-signal. $(\tau = 60,\ r = 13)$

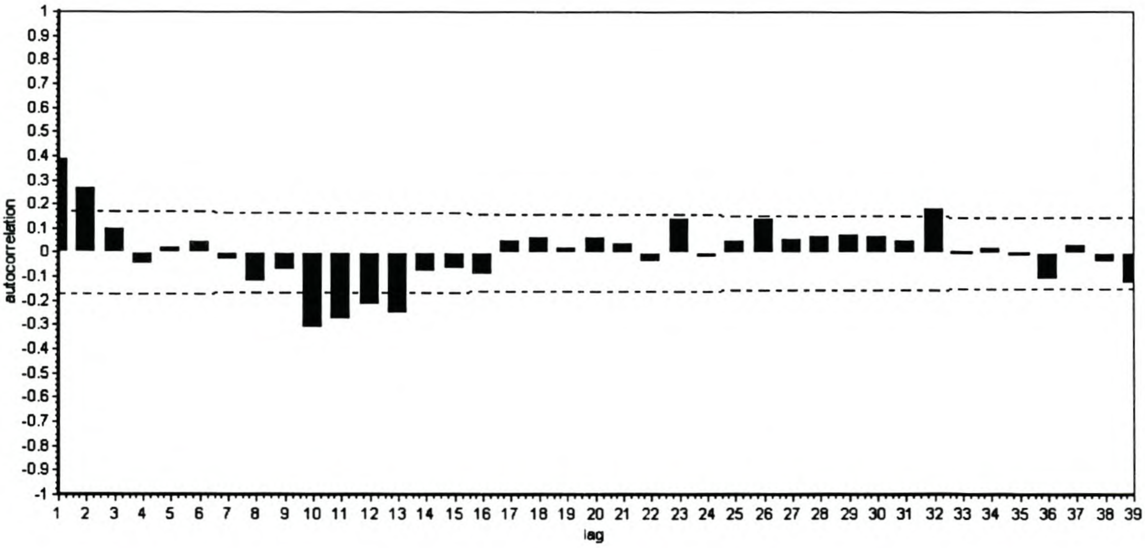


Figure 4.14 Sample autocorrelation function (SACF) of residual series.

The sample autocorrelation function (SACF) in Figure 4.14, above, was constructed using the residual series, which is exhibited in Figure 4.13. The confidence bands in the SACF represent 95% confidence for a null-hypothesis of white noise residuals. If trend was present in the residual series, then the SACF would be dying down extremely slowly towards zero. This is not the case. If any form of periodicity was left in the residual series, then the SACF would also exhibit a cyclical variation. This is also not the case. It would therefore seem that 13 leading eigenvectors have captured all the features in the series that SSA can handle.

Example 4.9 Forward validation model selection in the case of the hotel series using 12***-period-ahead forecasting***

The hotel series was used for the purposes of this example. The series was not transformed and $k = 12$-period-ahead forecasting was used during FV. Recall that this series is of length $N = 168$. The series was split into three sets, i.e. training set $\{x_t\}_{t=1}^{133}$, validation set $\{x_t\}_{t=145}^{156}$ and test set $\{x_t\}_{t=157}^{168}$. The maximum number of leading eigenvectors used was set at $r_{\max} = 13$. The window length looped over during FV was $25 \leq \tau \leq 67$. The FV analysis used 13 minutes and 18 seconds of CPU (800mHz) time.

The FV results are summarised in Table 4.9 and Figure 4.15 below. The best candidate model used 12 leading eigenvectors of this time series.

Table 4.9 Forward validation results for the hotel series when 12-period-ahead forecasting is used.

Number of leading eigenvectors (r)	FV_r (bias^2 + variance)	Mean(MSE) (bias)	Var(MSE) (variance)
12	593763.523	770.236	500.023
10	1052277.009	1025.469	690.134
11	1139698.043	1054.701	27304.844
8	1857306.026	1362.585	668.026
9	1860149.990	1363.479	1076.236
7	4173599.349	2042.712	925.744
6	9143518.906	3023.643	1100.758
5	9205395.157	3033.897	864.443
4	29610797.248	5441.470	1201.444
3	29924034.386	5470.210	834.854
13	1846297709.774	38972.945	327407257.301
2	5195618667637.230	519424.242	4925817124775.590
1	481184026493673.000	8145906.420	414828235082938.000

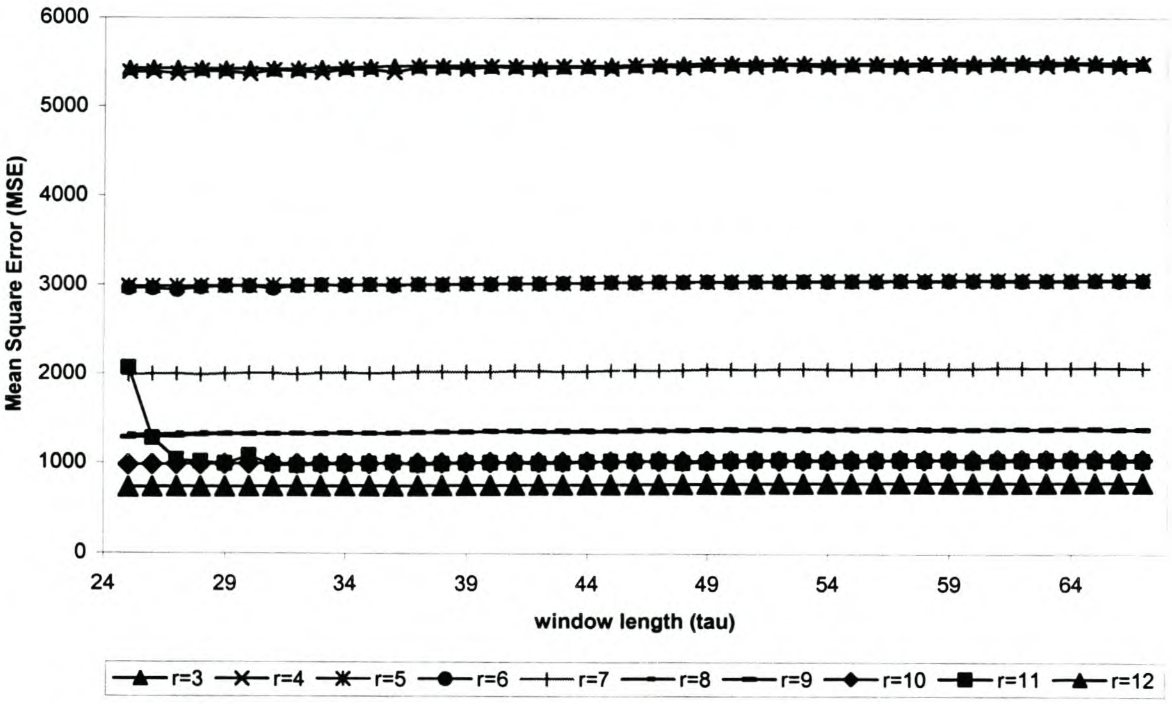


Figure 4.15 Forward validation results of the hotel series.

Figure 4.16, below, exhibits the first 156 observations of the hotel series and the reconstructed Cadzow-signal series. The series contains a monthly variation. This is also the highest period length, i.e. 12, in this series. A window length of $\tau = 72$ and the 12 leading eigenvectors were therefore used when constructing the Cadzow-signal series.

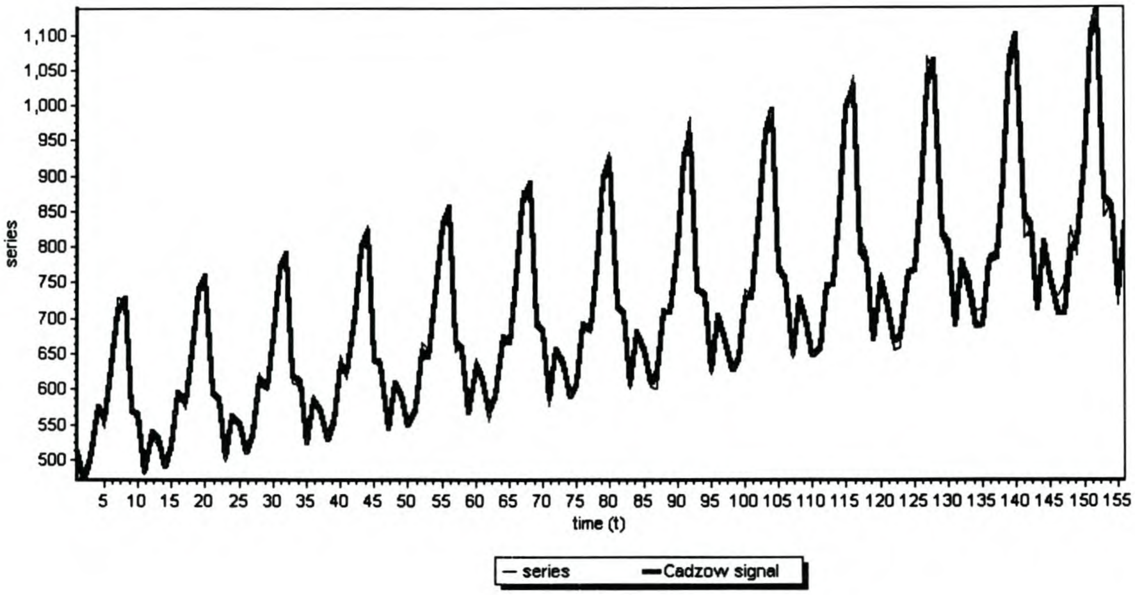


Figure 4.16 Hotel series and reconstructed Cadzow-signal series. $(\tau = 72,\ r = 12)$

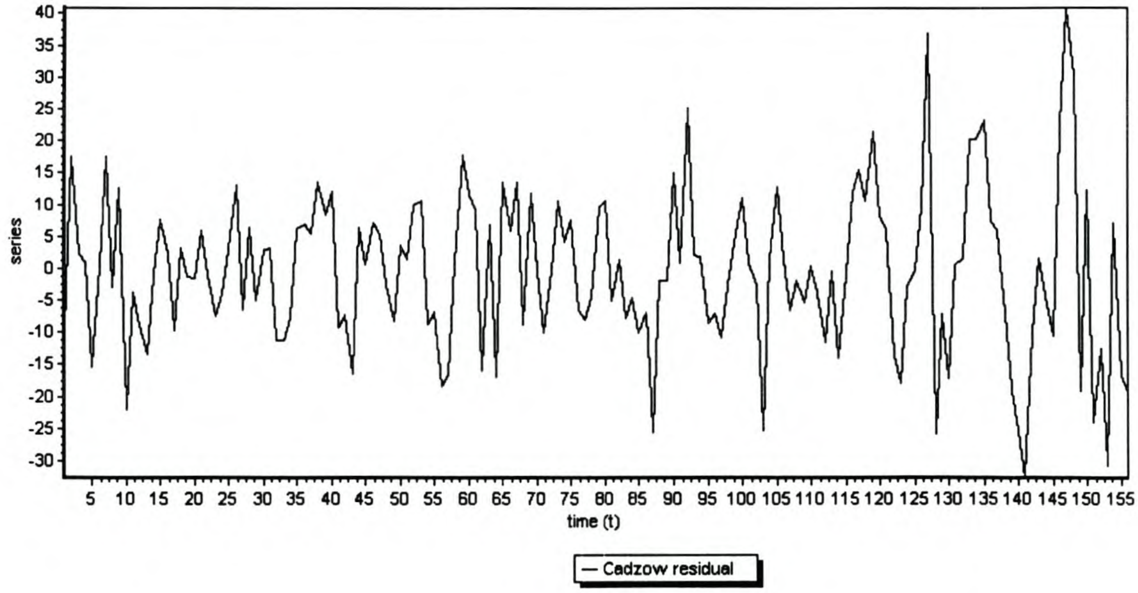


Figure 4.17 Residual series of the hotel series and Cadzow-signal. $(\tau = 72,\ r = 12)$

The residual series, exhibited in Figure 4.17, above, evidently contains no residual trend or cyclical variation. It would therefore seem that the FV model selection result, of 12 leading eigenvectors, captured all the features in the series that SSA can handle. The SACF below supports this argument and also that the residual series supports a white noise hypothesis.

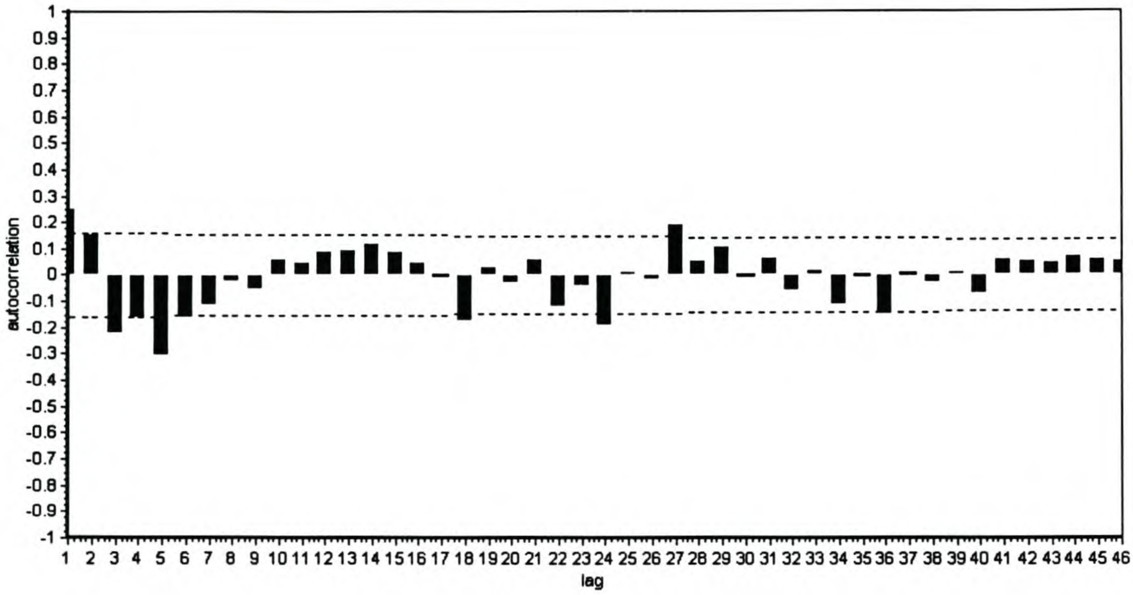


Figure 4.18 Sample autocorrelation function (SACF) of residual series.

To test the out-of-sample forecasting accuracy, a window length of $\tau = 72$ and the 12 leading eigenvectors were used. The results are summarized in Table 4.10 and Table 4.11.

Table 4.10 Recurrent one-period-ahead forecasting results when no transformation, window length $\tau = 72$***, centring of trajectory matrix and*** 12 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	17.24120	1.94%	383.03945
Approximate series	19.66284	2.19%	485.36494
Cadzow-signal	21.92186	2.43%	644.89919

Table 4.11** 12 **-period-ahead forecasting results when no transformation, window length** $\tau = 72$**, centring of trajectory matrix and** 12 **leading eigenvectors are used.

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	17.68150	1.97%	402.02949
Approximate series	20.26817	2.25%	523.71941
Cadzow-signal	21.92085	2.43%	644.91053
Vector forecasting	19.51400	2.20%	500.91110

Example 4.10 Forward validation model selection in the case of the liquor series using** 12 **-period-ahead forecasting

The liquor series was used for the purposes of this example. The series was log-transformed and $k = 12$ period-ahead-forecasting was used during FV. The series was split into three sets, i.e. training set $\{x_t\}_{t=1}^{301}$, validation set $\{x_t\}_{t=313}^{324}$ and test set $\{x_t\}_{t=325}^{336}$. The maximum number of leading eigenvectors used was set at $r_{\max} = 13$. The window length looped over during FV was $25 \le \tau \le 80$, even though the maximum window length could be set at $\tau_{\max} = 151$. The FV analysis used 3 hours, 34 minutes and 45 seconds of CPU (800mHz) time.

The FV results are summarised in Table 4.12 and Figure 4.19 below.

According to the sorted FV results in Table 4.12, the best candidate model used 13 leading eigenvectors.

Table 4.12 Forward validation results for the liquor series when using 12 *-period-ahead forecasting.*

Number of leading eigenvectors (r)	FV$_r$ (bias^2 + variance)	Mean(MSE) (bias)	Var(MSE) (variance)
13	112082802.251	10586.834	1742.823
12	299952900.166	17319.055	3218.083
10	372287158.872	19294.615	4991.194
11	459688287.529	21440.309	1444.366
8	610122285.488	24700.567	4274.174
9	747035122.050	27331.917	1418.441
7	1076907113.509	32816.243	1300.541
6	1316928603.831	36289.447	4633.251
5	1894602470.746	43527.016	1308.369
4	2511508704.906	50114.909	4557.198
3	3184445641.125	56430.882	1252.165
2	4329041413.948	65795.419	4252.812
1	5089905784.552	71343.567	1177.469

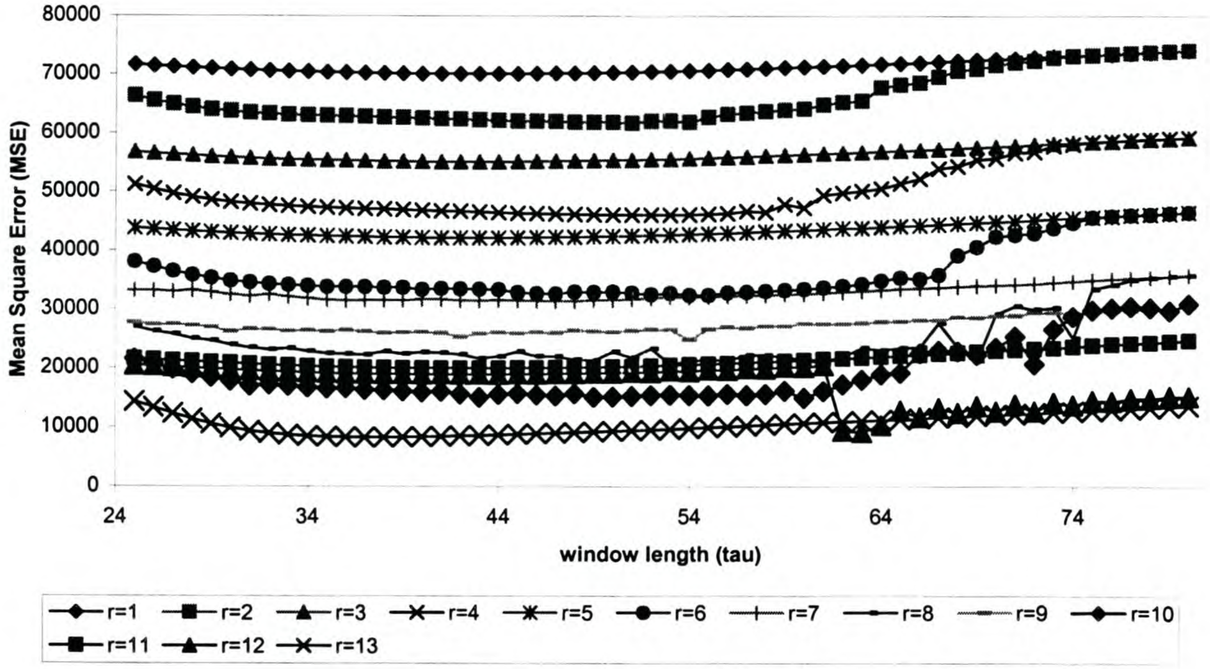


Figure 4.19 Forward validation results of the liquor series.

The first 324 observations of the liquor series and the reconstructed Cadzow-signal series for this series are exhibited in Figure 4.20, below. A window length of $\tau = 156$ (multiple of period length 12) and the 13 leading eigenvectors were used.

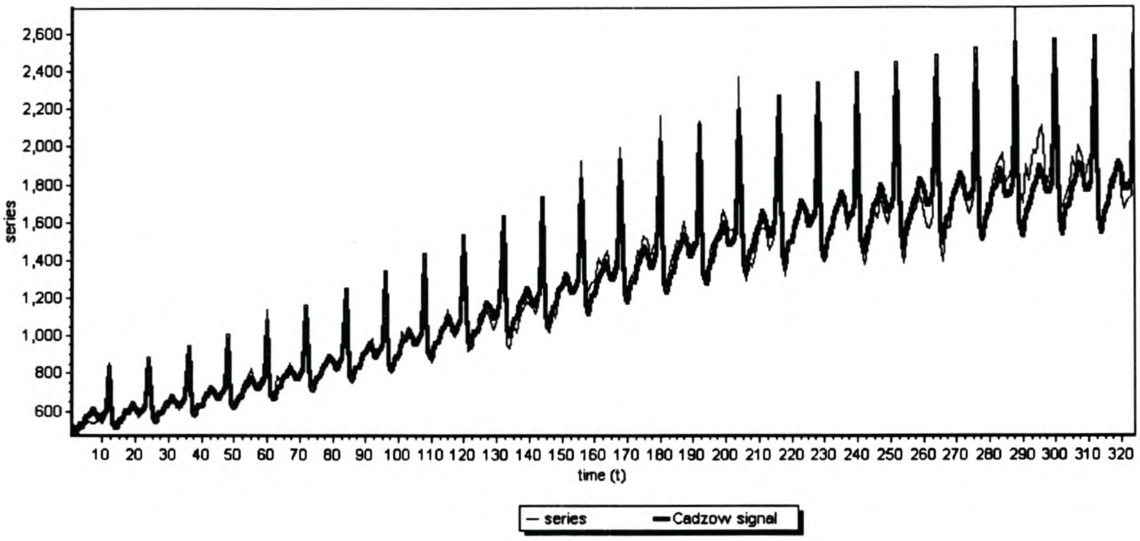


Figure 4.20 Liquor series and reconstructed Cadzow-signal series. $(\tau = 156,\ r = 13)$

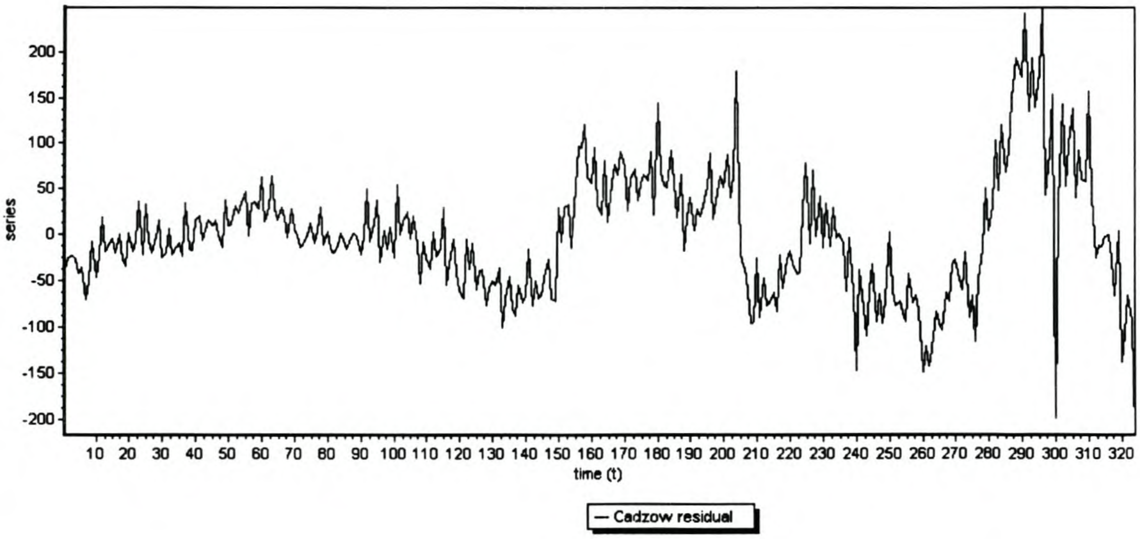


Figure 4.21 Residual series of the liquor series and Cadzow-signal. $(\tau = 156,\ r = 13)$

A few disturbing features can be deduced from perusal of Figure 4.21 above and Figure 4.22 below. The residual series does not represent the features of a white noise series. There seems to be an autoregressive structure left in this residual series and is supported from perusal of the SACF below. Another alarming feature is the sudden drop in the residual of observation 130. It seems plausible that this observation is possibly an additive outlier. This might lead to erroneous forecasting, due to the possible inclusion of such an observation in the vector, used in the forecasting algorithms presented in Chapter 3.

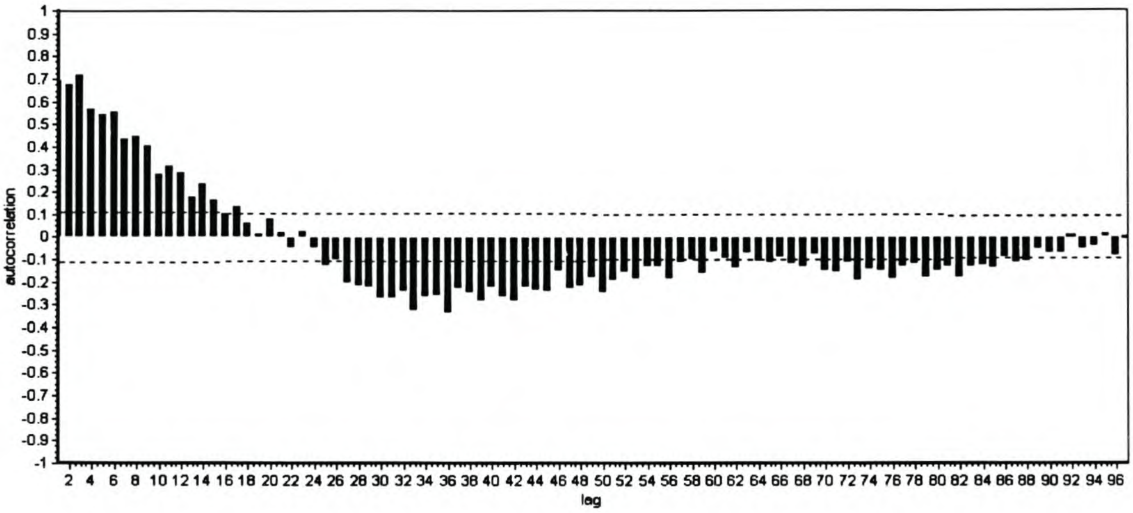


Figure 4.22 Sample autocorrelation function (SACF) of residual series.

To test out-of-sample forecasting accuracy, a window length of $\tau = 156$ and the 13 leading eigenvectors was used. The results are summarised in Table 4.13 and Table 4.14, below.

Table 4.13 Recurrent one-period-ahead forecasting results when a log-transformation, window length $\tau = 156$***, centring of trajectory matrix and*** 13 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	140.06990	8.12%	21034.36222
Approximate series	146.11794	8.40%	23233.58814
Cadzow-signal	74.39846	4.26%	7426.95094

Table 4.14 12***-period-ahead forecasting results when a log-transformation, window length*** $\tau = 156$***, centring of trajectory matrix and*** 13 ***leading eigenvectors are used.***

	Measures of forecasting accuracy		
Series used	**MAD**	**MAPE**	**MSE**
Original series	147.23297	8.56%	23248.99787
Approximate series	157.33433	9.09%	26712.05208
Cadzow-signal	74.37813	4.26%	7424.17006
Vector forecasting	230.51525	13.14%	56441.11742

Remark 4.2

Algorithm 4.2 uses a certain number of leading eigenvectors during the forward validation model selection process. The selection of leading eigenvectors can possibly result in the selection of a number of leading eigenvectors, erroneously excluding the second eigenvector of an eigenvector pairing, which might jointly be used to explain periodic variation in a time series. To overcome such a situation an alternative algorithm to the above is also proposed in this study.

The following algorithm combines forward validation model selection with a bias variance trade-off together with the "grouping" idea, which was introduced by Golyandina *et al.* (2001). In this procedure, artificial phase space portraits are constructed and groupings of eigenvectors are identified. This is done before the actual FV model selection stage. A model selection technique of this nature can, unfortunately, succumb to subjectivity brought in by the actual choice of pairings of eigenvectors. Instead of looping over the number of leading eigenvectors, the new proposed algorithm loops over groups of eigenvectors. With each loop in the algorithm the previous group of eigenvectors is increased with a single group of eigenvectors. This causes a sequential effect in the loop structure. The groups of eigenvectors, constructed in this algorithm, honour the importance of eigenvectors in the sense of percentage variation explained by individual eigenvectors. It will, for this reason, be seen that Algorithm 4.2 and Algorithm 4.3 produces very similar results.

Algorithm 4.3 Sequential grouping forward validation with k-period-ahead forecasting

(a) Load an observed time series $\{x_t\}_{t=1}^{N}$ and transform if necessary.

(b) Set the number of periods to forecast ahead (k). Split the time series in (a) into a training set $\{x_t\}_{t=1}^{N_1-k+1}$, validation set $\{x_t\}_{t=N_1+1}^{N_1+N_2}$ and test set $\{x_t\}_{t=N_1+N_2+1}^{N}$.

(c) Construct an artificial phase space portrait based on eigenvectors obtained through SVD of the centred trajectory matrix of the training set. Note that eigenvectors are ordered according to their individual percentage variation explained in descending order. Identify individual eigenvectors that are slowly varying and possibly explain trend. Identify pairs of consecutive eigenvectors that possibly explain cyclical patterns. Set up a grouping matrix $\mathbf{C}$ containing three columns as follows

$$\underset{p\times 3}{\mathbf{C}} = \begin{bmatrix} \overbrace{c_{1,1}}^{\text{Number of Eigenvectors in group}} & \overbrace{c_{1,2}}^{\text{Index 1}} & \overbrace{c_{1,3}}^{\text{Index 2}} \\ \vdots & \vdots & \vdots \\ c_{p,1} & c_{p,2} & c_{p,3} \end{bmatrix}. \tag{4.27}$$

The first column contains the number of eigenvectors per group and will either be 1 for a single eigenvector that explains trend or 2 for a pair of eigenvectors that explain a cyclical pattern. The second column will contain the index value of the eigenvector that explains trend. The second and third column vectors will jointly contain the index values of the pair of eigenvectors that explain a cyclical pattern. The total number of eigenvector groupings is equal to p, which is equal to the number of rows in the grouping matrix.

(d) Set $r_{\max}$ equal to the total number of eigenvectors in matrix $\mathbf{C}$, i.e. let $r_{\max} = \sum_{i=1}^{p} c_{i,1}$ and loop over the window length (τ), where $k + r_{\max} \leq \tau \leq \tau_{\max}(\leq \text{integer part of } (N_1 - k + 2)/2)$.

(e) Start a loop that loops over the number of rows in matrix $\mathbf{C}$, i.e. $1 \leq k \leq p$. Then set the number of eigenvectors to use equal to $r = \sum_{i=1}^{k} c_{i,1}$.

(f) Set up the index set $I = \left\{c_{1,2}, c_{1,3}, \ldots, c_{k,2}, c_{k,3}\right\}$ containing the index values of the r eigenvectors to use. Use this index set of eigenvectors and window length in step (d) to construct a Cadzow-signal series.

(g) Unfold the Cadzow-signal series into the column vectors of a $\tau \times (N_1 - k - \tau + 2)$ trajectory matrix. Construct the centred trajectory matrix $\tilde{\mathbf{X}}$ and scatter matrix $\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}'\right)$.

(h) Perform SVD of the scatter matrix.

(i) Use the r eigenvectors in step (f) to construct the r column vectors of the matrix $\mathbf{V} = \left[\mathbf{v}_{c_{1,2}} \quad \mathbf{v}_{c_{1,3}} \quad \cdots \quad \mathbf{v}_{c_{k,2}} \quad \mathbf{v}_{c_{k,3}}\right]$. If $c_{i,3}$ has no value, i.e. no index referring to an eigenvector since the group contains only a single eigenvector grouping, then it is simply ignored when creating the latter matrix.

(j) Construct the projection matrix $\mathbf{P}_{\mathfrak{L}_r} = \mathbf{V}\mathbf{V}'$.

(k) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r^\perp} = \mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}$.

(l) Partition the projection matrix in (k) as follows

$$\begin{aligned}\mathbf{P}_{\mathfrak{L}_r^\perp} &= \left[\mathbf{P}_{\mathfrak{L}_r^\perp}\left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-k}\right] \,\middle|\, \mathbf{P}_{\mathfrak{L}_r^\perp}\left[\mathbf{e}_{\tau-k+1} \quad \cdots \quad \mathbf{e}_\tau\right]\right] \\ &= \left[\left.\underset{\tau\times(\tau-k)}{\mathbf{P}_1}\,\right|\,\underset{\tau\times k}{\mathbf{P}_2}\right],\end{aligned} \tag{4.28}$$

where $\mathbf{e}_j$ is the j^{th} column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and k is the number of periods being forecast ahead.

(m)Calculate the k-period-ahead forecasts using

$$\begin{bmatrix} \hat{x}_{N_1-k+2} \\ \vdots \\ \hat{x}_{N_1+1} \end{bmatrix} = \begin{bmatrix} \overline{x}_{\tau-k+1} \\ \vdots \\ \overline{x}_\tau \end{bmatrix} - \left(\mathbf{P}_2'\mathbf{P}_2\right)^{-1}\left(\mathbf{P}_2'\mathbf{P}_1\right)\begin{bmatrix} x_{N_1-\tau+2} - \overline{x}_1 \\ \vdots \\ x_{N_1-k+1} - \overline{x}_{\tau-k} \end{bmatrix}, \tag{4.29}$$

where

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{P}_{\mathfrak{L}_r^{\perp}} \left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-k} \right] \\ \mathbf{P}_2 &= \mathbf{P}_{\mathfrak{L}_r^{\perp}} \left[\mathbf{e}_{\tau-k+1} \quad \cdots \quad \mathbf{e}_{\tau} \right] . \end{aligned} \tag{4.30}$$

(n) If a transformation was applied to the series in step (a) then a back-transformation must be used in this step. Calculate the mean squared prediction error $\frac{1}{k}\sum_{i=1}^{k}\left(x_{N_1-k+i+1} - \hat{x}_{N_1-k+i+1}\right)^2$. Increase the training set length to $N_1 = N_1 + 1$ and return to step (f). The process is continued until the whole validation set has been forecast k-periods-ahead.

(o) Form the following matrix

$$\underset{(\tau_{\max}-k-r_{\max}+1)\times p}{\mathbf{MSE}} = \begin{bmatrix} mse_{k+r_{\max},1} & mse_{k+r_{\max},2} & \cdots & mse_{k+r_{\max},p} \\ mse_{k+r_{\max}+1,1} & mse_{k+r_{\max}+1,2} & \cdots & mse_{k+r_{\max}+1,p} \\ \vdots & \vdots & \vdots & \vdots \\ mse_{\tau_{\max},1} & mse_{\tau_{\max},2} & \cdots & mse_{\tau_{\max},p} \end{bmatrix}, \tag{4.31}$$

where

$$mse_{i,j} = \frac{1}{kN_2} \sum_{t=N_1+1}^{N_1+N_2} \sum_{i=1}^{k} \left(x_{t-k+i} - \hat{x}_{t-k+i} \right)^2 \tag{4.32}$$

for $k + r_{\max} \le i \le \tau_{\max}$ and $1 \le j \le p$. Repeat until loops (e) and (d) have been completed.

(p) Calculate the following measure

$$FV_j = \underbrace{\left[\frac{1}{\tau_{\max} - r_{\max} - k + 1}\sum_{i=k+r_{\max}}^{\tau_{\max}} mse_{i,j}\right]^2}_{\text{mean(MSE)}^2}$$

$$+\underbrace{\frac{1}{\tau_{\max} - r_{\max} - k}\sum_{i=k+r_{\max}}^{\tau_{\max}}\left[mse_{i,j} - \left\{\frac{1}{\tau_{\max} - r_{\max} - k + 1}\sum_{i=k+r_{\max}}^{\tau_{\max}} mse_{i,j}\right\}\right]^2}_{\text{var(MSE)}}$$

(4.33)

for $1 \leq j \leq p$.

Remark 4.3

(a) The above algorithm incorporates the idea of "grouping" eigenvectors, according to certain structures that they possess (Golyandina *et al.*, 2001), and also FV model selection. It is felt that grouping is a relatively subjective issue and also causes a model selection technique not to be automatic. The actual FV model selection process is superseded by a visual inspection of artificial phase space portraits for the purposes of "grouping" eigenvectors.

(b) The grouping of eigenvectors, used in step (c), is not arbitrary. The percentage variation explained by eigenvectors, are taken into consideration when setting up the grouping matrix $\mathbf{C}$. The groups are formed using the features they possess in the artificial phase space portrait (slowly varying or "star"-like patterns), which are based on eigenvectors that were obtained through SVD, honouring the percentage variation explained by each eigenvector. The pairings of eigenvectors, found in the grouping matrix, are therefore sorted in accordance with the percentage variation explained by the eigenvectors, and appear in descending order in the rows of this matrix.

(c) The $\mathbf{MSE}$ matrix in step (n) contains p row vectors. Each of these rows represents a candidate model, which sequentially contains more pairings of eigenvectors. The first column vector corresponds to the eigenvector pairing $\left\{\mathbf{v}_{c_{1,2}}, \mathbf{v}_{c_{1,3}}\right\}$, where $c_{1,2}$ is the index value of the first eigenvector to use. If $c_{1,3}$ has no index value, then it should be clear that the pairing will simply be $\left\{\mathbf{v}_{c_{1,2}}\right\}$. The second column vector corresponds to the eigenvector pairing $\left\{\mathbf{v}_{c_{1.2}}, \mathbf{v}_{c_{1.3}}\right\}, \left\{\mathbf{v}_{c_{2.2}}, \mathbf{v}_{c_{2.3}}\right\}$. The final candidate model consists of the eigenvector pairing $\left\{\mathbf{v}_{c_{1.2}}, \mathbf{v}_{c_{1.3}}\right\}, \left\{\mathbf{v}_{c_{2.2}}, \mathbf{v}_{c_{2.3}}\right\}, \ldots, \left\{\mathbf{v}_{c_{p.2}}, \mathbf{v}_{c_{p.3}}\right\}$.

(d) This algorithm is virtually the same as Algorithm 4.2 and only differs in the manner in which eigenvectors are grouped and used in a sequential fashion to produce candidate models.

Example 4.11 Sequential grouping forward validation in the case of the airline time series

A log-transformation was applied to the airline passenger series. The number of periods being forecast ahead was set at $k = 12$. The airline series was then split into three sets, i.e. training set $\left\{x_t\right\}_{t=1}^{109}$, validation set $\left\{x_t\right\}_{t=121}^{132}$ and test set $\left\{x_t\right\}_{t=133}^{144}$.

Figure 4.23, below, exhibits the artificial phase space portraits that were constructed using the log-transformed airline passenger time series. The graphs represents plots of paired eigenvectors and are used to search for pairs of eigenvectors that explain periodic variation in a time series. The pairs of eigenvectors $\{2,3\}$, $\{4,5\}$, $\{8,9\}$, $\{10,11\}$ and $\{12,13\}$ seem to fit within the framework introduced in Section 4.4.3.

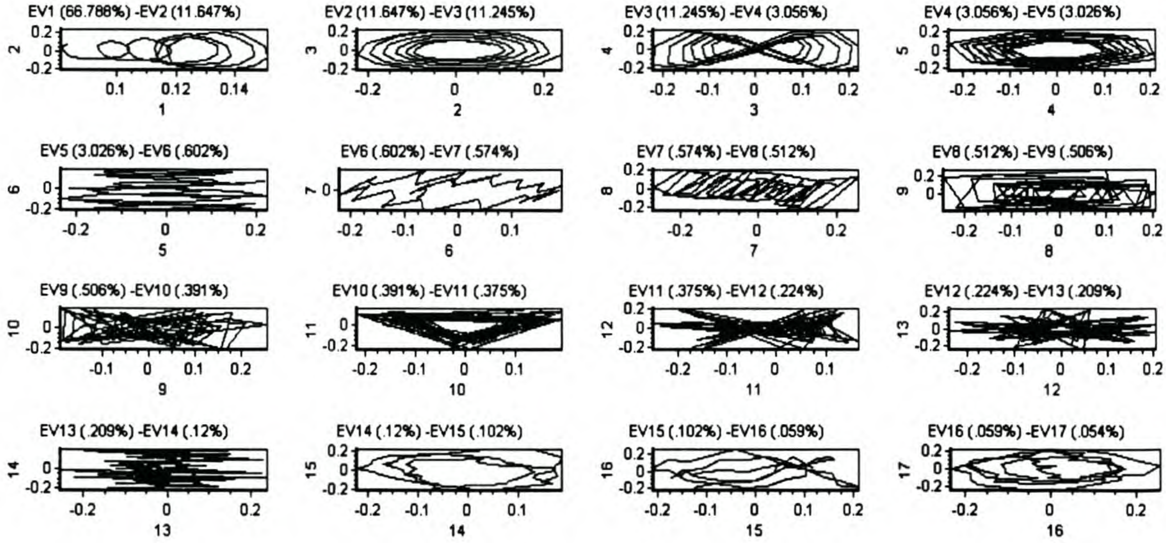


Figure 4.23 Artificial phase space portrait of log-transformed airline series. (paired eigenvectors)

Figure 4.24, below, exhibits the artificial phase space portraits, which were constructed using individual eigenvectors. This representation is used to search for eigenvectors, which vary slowly. The eigenvectors $\{1\},\{6\}$ and $\{7\}$ seem to conform to this description.

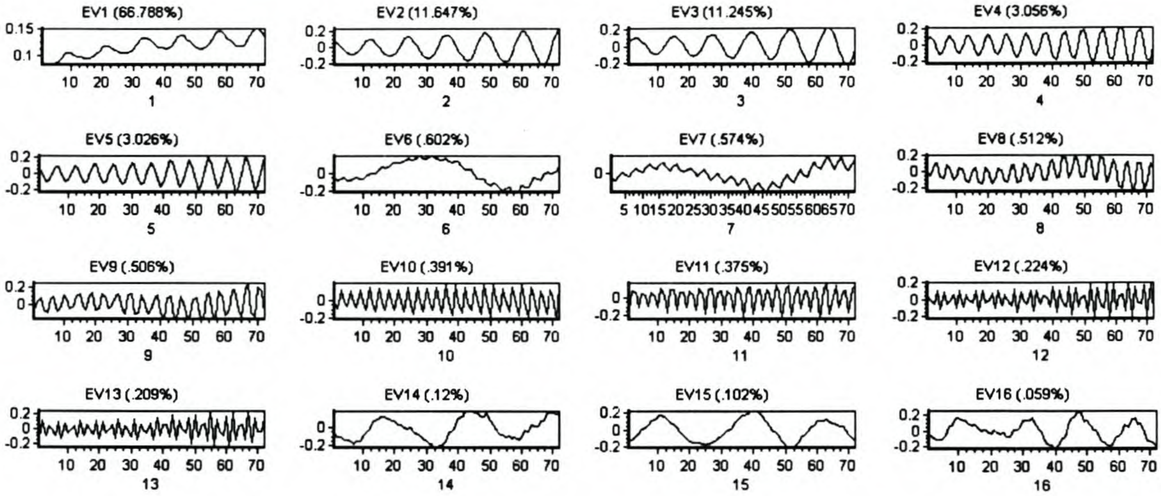


Figure 4.24 Artificial phase space portrait of log-transformed airline series. (individual eigenvectors)

The following groups of eigenvectors were therefore identified in using the artificial phase space portraits,

$$\left\{\overbrace{1}^{trend},\ \underbrace{\overbrace{(2,3)}^{cycle}}_{period=12},\ \underbrace{\overbrace{(4,5)}^{cycle}}_{period=6},\ \overbrace{6}^{trend},\ \overbrace{7}^{trend},\ \underbrace{\overbrace{(8,9)}^{cycle}}_{period=4},\ \underbrace{\overbrace{(10,11)}^{cycle}}_{period=3},\ \underbrace{\overbrace{(12,13)}^{cycle}}_{period=?}\right\}. \tag{4.34}$$

The information in (4.34) was then used to set up the following grouping matrix,

$$\underset{8\times3}{C} = \begin{bmatrix} 1 & 1 & \\ 2 & 2 & 3 \\ 2 & 4 & 5 \\ 1 & 6 & \\ 1 & 7 & \\ 2 & 8 & 9 \\ 2 & 10 & 11 \\ 2 & 12 & 13 \end{bmatrix}. \tag{4.35}$$

The FV analysis used 3 minutes and 22 seconds of CPU (800mHz) time. The results are summarised in Table 4.15 and Figure 4.25 below.

Table 4.15 FV results for airline series when using* 12 *-period-ahead forecasting.

Grouping of Eigenvectors (model)	FV_{model} (bias^2 + variance)	Mean(MSE) (bias)	Var(MSE) (variance)
{1},{2,3},{4,5},{6},{7},{8,9},{10,11},{12,13}	177703.915	408.848	10547.230
{1},{2,3},{4,5},{6},{7},{8,9},{10,11}	274478.271	514.158	10119.815
{1},{2,3},{4,5},{6},{7},{8,9}	407892.611	634.141	5757.214
{1},{2,3},{4,5},{6},{7}	845855.553	871.716	85967.108
{1},{2,3},{4,5}	971820.470	985.444	719.789
{1},{2,3}	2546006.103	1595.227	1257.750
{1},{2,3},{4,5},{6}	18357639.829	4170.098	967924.900
{1}	21957424.538	4685.717	1476.106

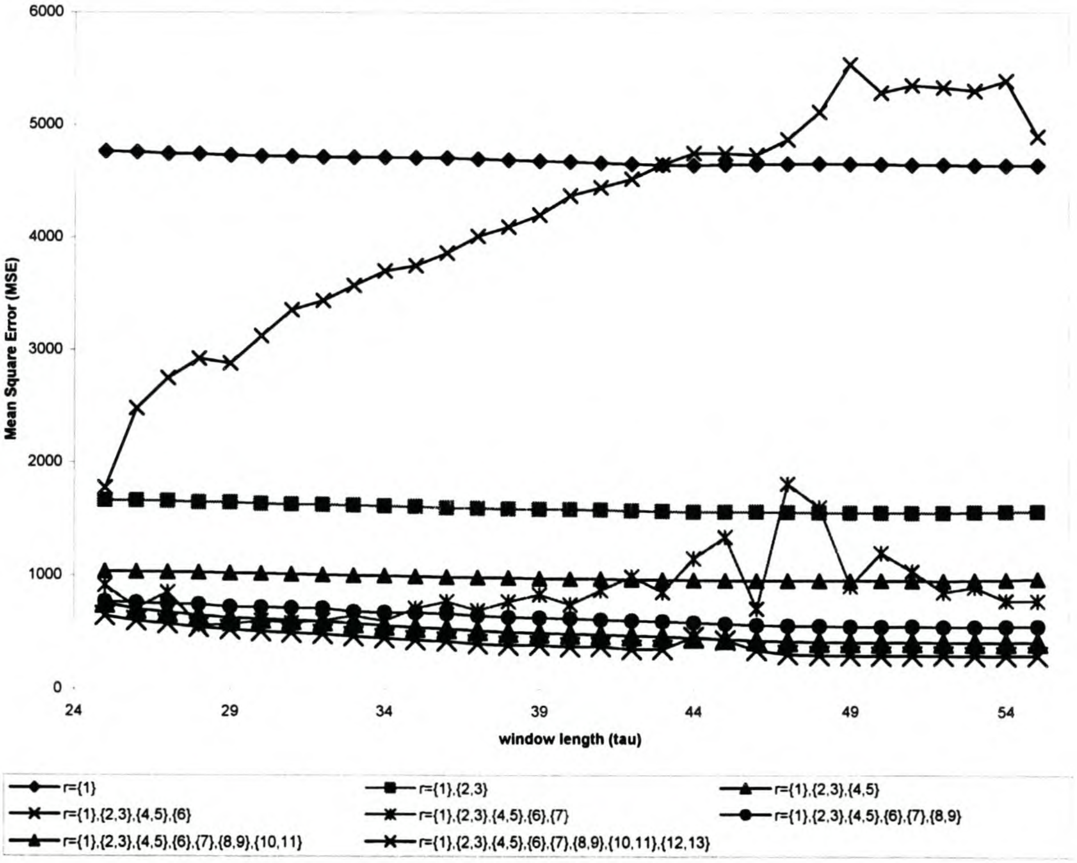


Figure 4.25 Sequential grouping forward validation results of the airline series.

According to the FV results in Table 4.15, above, the top model has the eigenvector pairings,

$$\{1\},\{2,3\},\{4,5\},\{6\},\{7\},\{8,9\},\{10,11\},\{12,13\}. \tag{4.36}$$

This "model" therefore uses the 13 leading eigenvectors. It is therefore clear that the FV method, introduced in Algorithm 4.2, produced the same result as the one in this example. If the results in Table 4.15 and Table 4.8 are compared, it follows that the top 5 models are exactly the same, i.e. using the 13, 11, 9, 7 and 5 leading eigenvectors.

Example 4.12 Sequential grouping forward validation in the case of the hotel time series

The hotel series was used for the purposes of this example. The series was not transformed and $k = 12$ period-ahead-forecasting was used during FV. The series was then split into three sets, i.e. training set $\left\{x_t\right\}_{t=1}^{133}$, validation set $\left\{x_t\right\}_{t=145}^{156}$ and test set $\left\{x_t\right\}_{t=157}^{168}$.

Figure 4.26 exhibits the artificial phase space portraits, which are all based on paired eigenvectors.

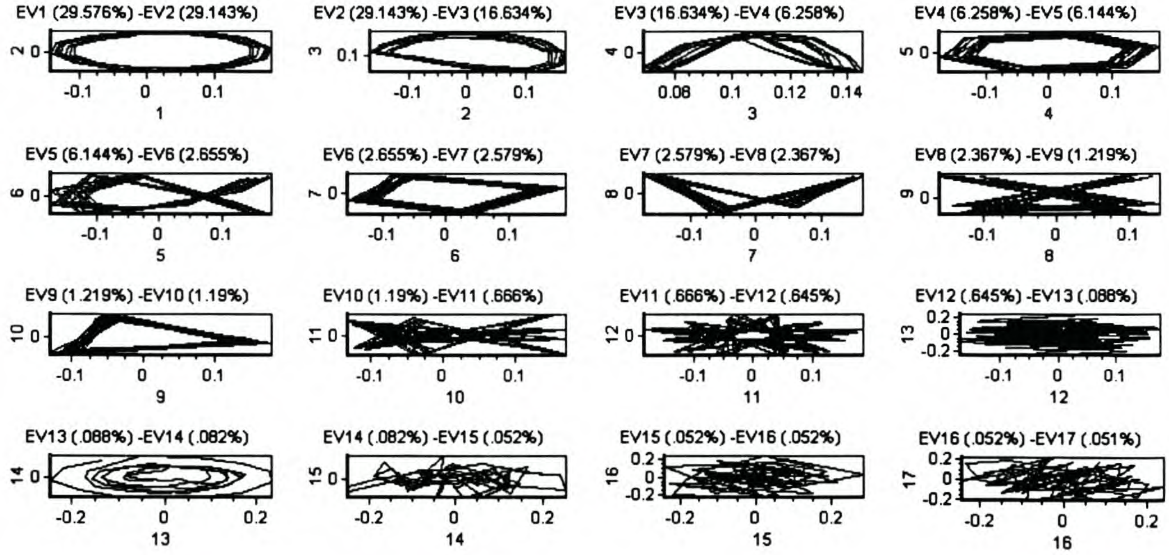


Figure 4.26 Artificial phase space portrait of the hotel series. (paired eigenvectors)

The pairs of eigenvectors $\{1,2\},\{4,5\},\{6,7\},\{9,10\}$ and $\{11,12\}$ were identified as having "star"-like shapes. The consecutive eigenvectors in these pairings also have percentage variation explained which are fairly similar.

Figure 4.27, below, exhibits phase space portraits of the individual eigenvectors. The eigenvectors $\{3\},\{8\}$ were identified as individual eigenvectors, amongst the leading eigenvectors, that describe pattern.

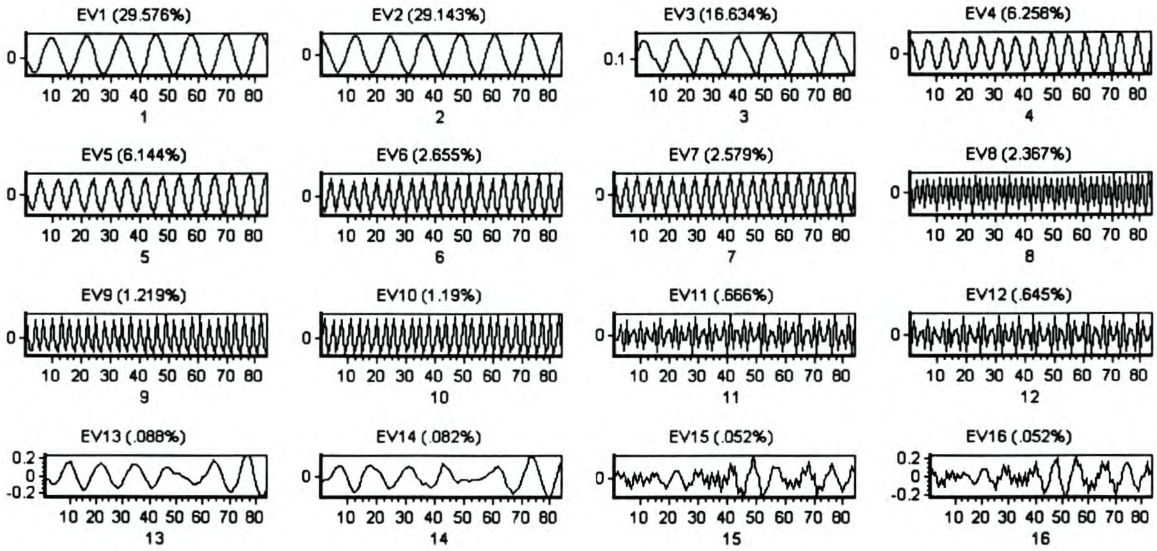


Figure 4.27 Artificial phase space portrait of the hotel series. (individual eigenvectors)

The following grouping of eigenvectors was then used during the FV procedure.

$$\left\{ \underbrace{\overbrace{(1,2)}^{cycle}}_{period=12}, 3, \underbrace{\overbrace{(4,5)}^{cycle}}_{period=6}, \underbrace{\overbrace{(6,7)}^{cycle}}_{period=4}, 8, \underbrace{\overbrace{(9,10)}^{cycle}}_{period=3}, \underbrace{\overbrace{(11,12)}^{cycle}}_{period=?} \right\} \tag{4.37}$$

The eigenvector pairings in (4.37) were used to set up the grouping matrix in (4.38), which was used in the forward validation procedure,

$$\underset{7\times 3}{C} = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 3 & \\ 2 & 4 & 5 \\ 2 & 6 & 7 \\ 1 & 8 & \\ 2 & 9 & 10 \\ 2 & 11 & 12 \end{bmatrix}. \qquad (4.38)$$

The forward validation analysis used 5 minutes 9 seconds of CPU (800mHz) time. Results of this analysis are exhibited in Table 4.16 and Figure 4.28 below.

The FV results in Table 4.16 can be compared to those summarised in Table 4.9 on page 198. The top two models in Table 4.16 respectively uses the 12 and 10 leading eigenvectors. Some minor differences in the selected models are then encountered. It is, however, very comforting that two examples thus far have shown that Algorithm 4.2 and Algorithm 4.3 produce virtually the same results.

Table 4.16 FV results for hotel series when using* 12 *-period-ahead forecasting.

Grouping of Eigenvectors (model)	**FV_{model} (bias^2 + variance)**	**Mean(MSE) (bias)**	**Var(MSE) (variance)**
{1,2},{3},{4,5},{6,7},{8},{9,10},{11,12}	593763.523	770.236	500.023
{1,2},{3},{4,5},{6,7},{8},{9,10}	1052277.009	1025.469	690.134
{1,2},{3},{4,5},{6,7},{8}	1857306.026	1362.585	668.026
{1,2},{3},{4,5},{6,7}	4173599.349	2042.712	925.744
{1,2},{3},{4,5}	9205395.157	3033.897	864.443
{1,2},{3}	29924034.386	5470.210	834.854
{1,2}	5195618667637.230	519424.242	4925817124775.590

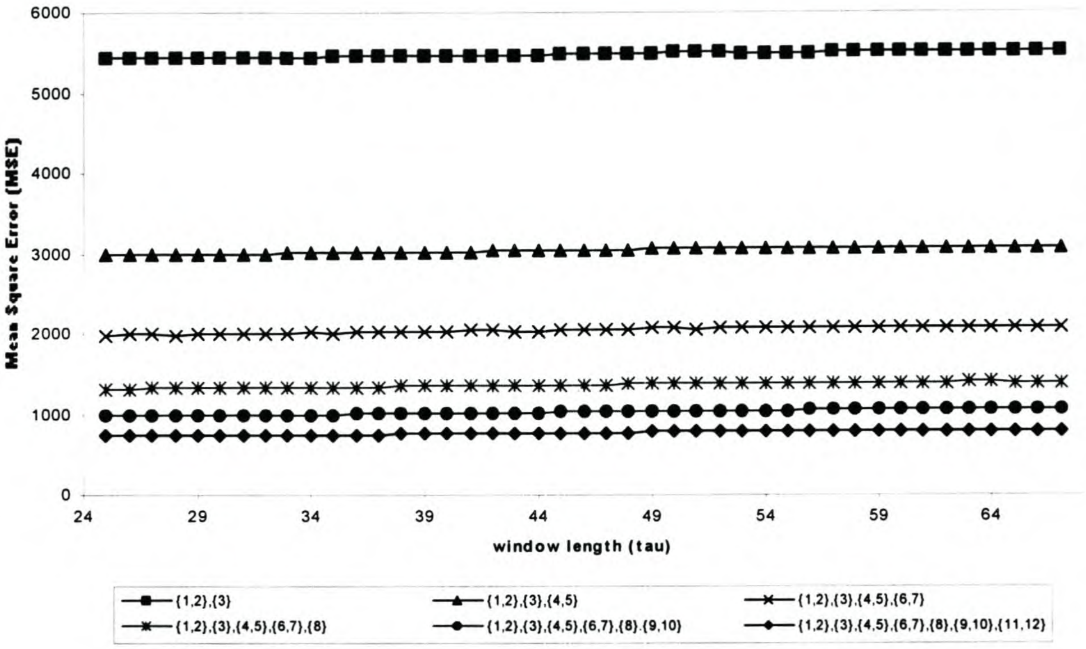


Figure 4.28 Sequential grouping forward validation results of the hotel series.

Example 4.13 Sequential grouping forward validation in the case of the liquor series

The liquor series was used for the purposes of this example. The series was log-transformed and $k = 12$ period-ahead-forecasting was used during FV. The series was split into three sets, i.e. training set $\left\{x_t\right\}_{t=1}^{301}$, validation set $\left\{x_t\right\}_{t=313}^{324}$ and test set $\left\{x_t\right\}_{t=325}^{336}$.

The artificial phase space portrait of the log-transformed series is exhibited in Figure 4.29, below. The pairs of eigenvectors $\{2,3\},\{4,5\},\{6,7\},\{8,9\}$ and $\{10,11\}$ were identified in this graphical representation.

The artificial phase space portraits in Figure 4.30 were used to identify slowly varying eigenvectors and individual eigenvectors that describe pattern. Eigenvectors $\{1\},\{12\}$ and $\{13\}$ were identified in this regard.

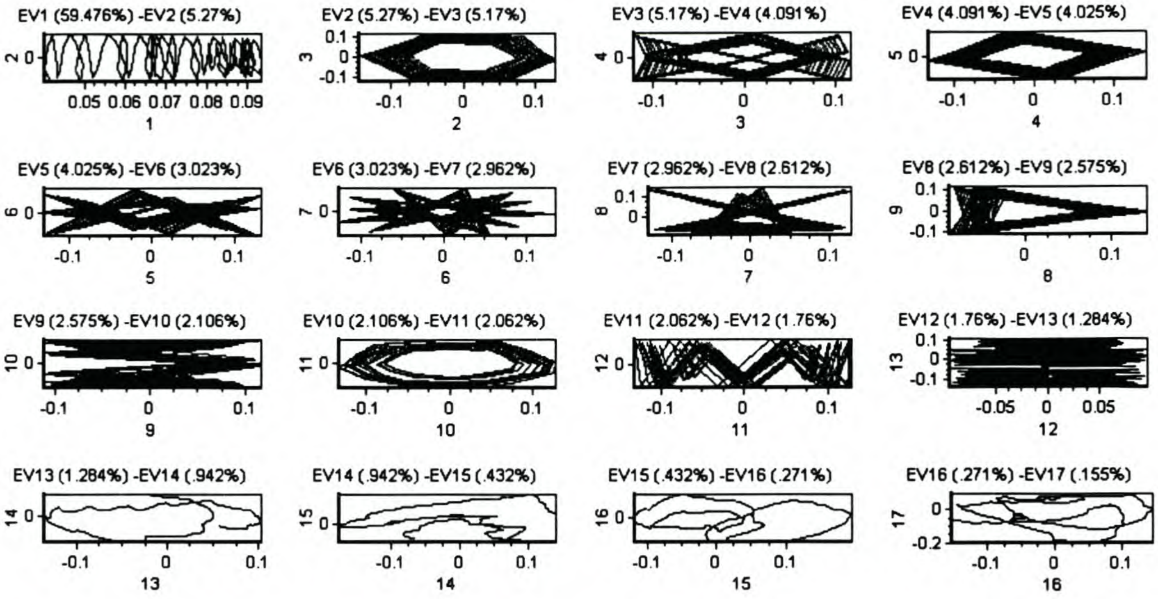


Figure 4.29 Artificial phase space portrait of the liquor series. (paired eigenvectors)

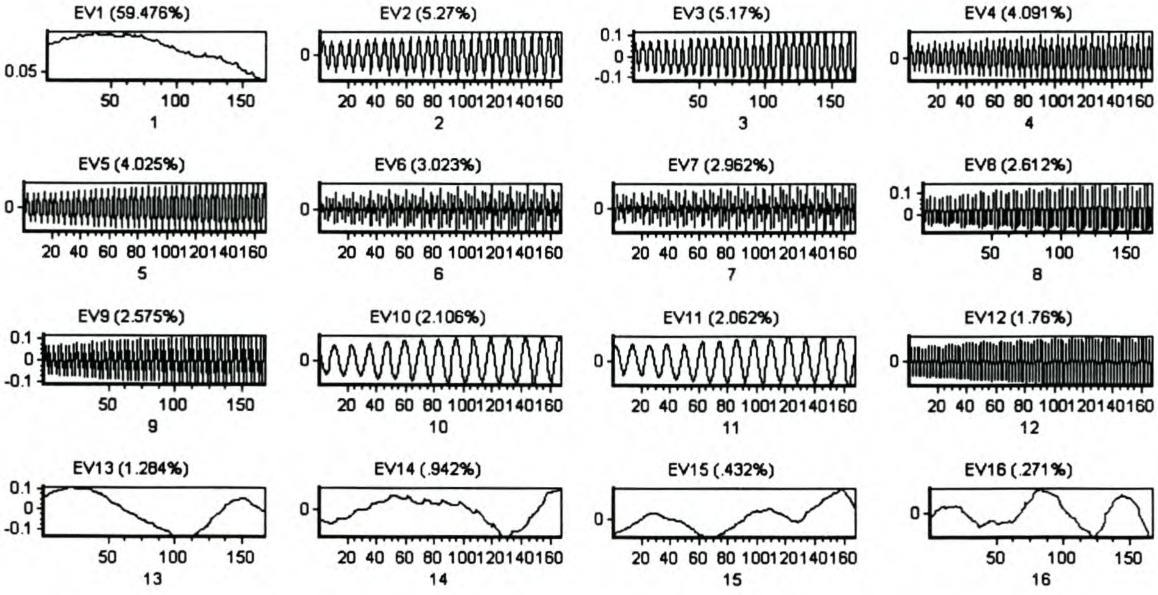


Figure 4.30 Artificial phase space portrait of the liquor series. (individual eigenvectors)

After visual inspection of the artificial phase space portraits, the following pairings of eigenvectors were identified

$$\left\{\overbrace{1}^{trend},\ \underbrace{\overbrace{(2,3)}^{cycle}}_{period=6},\ \underbrace{\overbrace{(4,5)}^{cycle}}_{period=4},\ \underbrace{\overbrace{(6,7)}^{cycle}}_{period=?},\ \underbrace{\overbrace{(8,9)}^{cycle}}_{period=3},\ \underbrace{\overbrace{(10,11)}^{cycle}}_{period=12},\ 12,\ 13\right\}. \qquad (4.39)$$

The pairings of eigenvectors in (4.39) were used to set up the grouping matrix in (4.40)

$$\underset{8\times 3}{C} = \begin{bmatrix} 1 & 1 & \\ 2 & 2 & 3 \\ 2 & 4 & 5 \\ 2 & 6 & 7 \\ 2 & 8 & 9 \\ 2 & 10 & 11 \\ 1 & 12 & \\ 1 & 13 & \end{bmatrix}. \qquad (4.40)$$

The FV analysis used 2 hours, 46 minutes and 35 seconds of CPU (800mHz) time. Results of this analysis are summarised in Table 4.17 and Figure 4.31 below.

Table 4.17 FV results for liquor series when using 12 ***-period-ahead forecasting.***

Grouping of Eigenvectors (model)	FV_{model} ($bias^2$ + variance)	Mean(MSE) (bias)	Var(MSE) (variance)
{1},{2,3},{4,5},{6,7},{8,9},{10,11},{12},{13}	10586.834	3037431.496	115118490.924
{1},{2,3},{4,5},{6,7},{8,9},{10,11},{12}	17319.055	10356061.182	310305743.265
{1},{2,3},{4,5},{6,7},{8,9},{10,11}	21440.309	2086193.121	461773036.284
{1},{2,3},{4,5},{6,7},{8,9}	27331.917	2011976.228	749045679.837
{1},{2,3},{4,5},{6,7}	32816.243	1691407.975	1078597220.943
{1},{2,3},{4,5}	43527.016	1711830.163	1896312992.540
{1},{2,3}	56430.882	1567916.402	3186012305.362
{1}	71343.567	1386432.637	5091291039.720

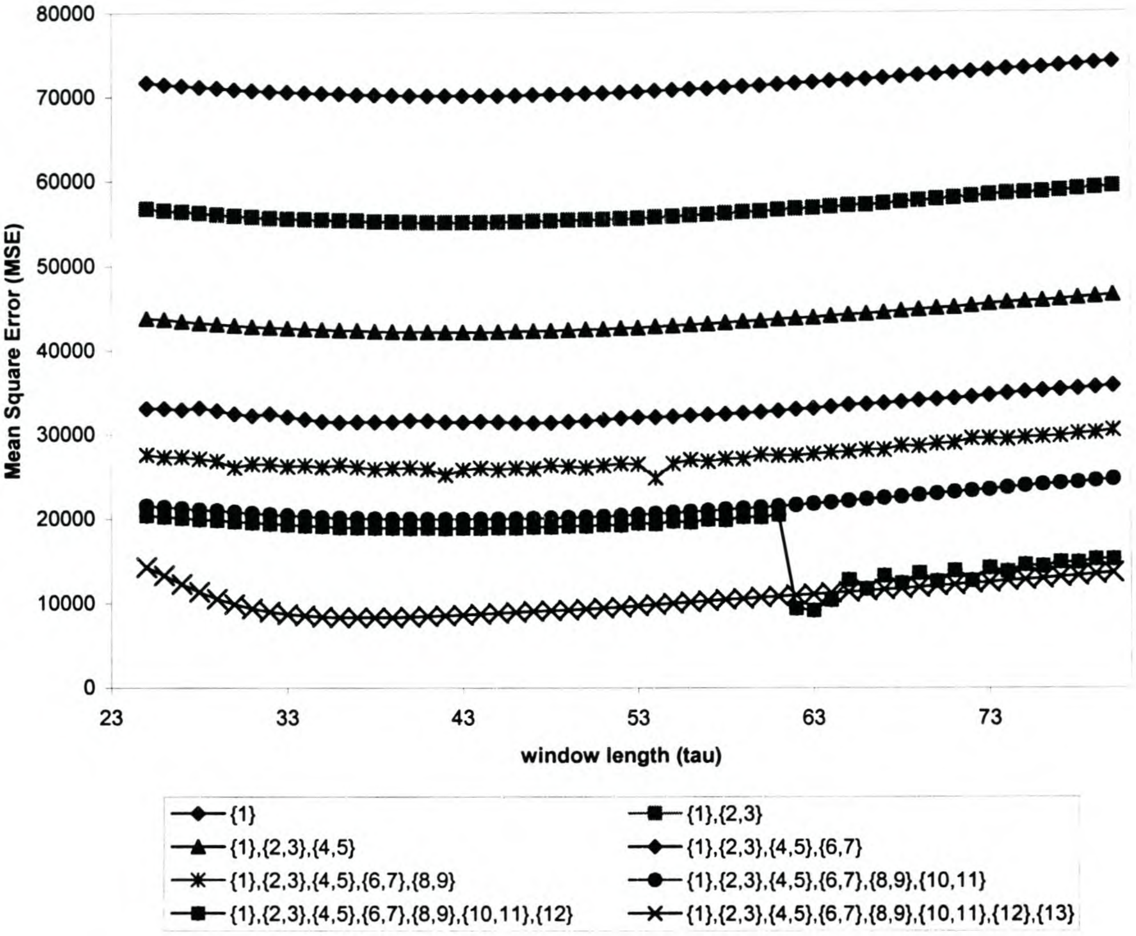


Figure 4.31 Sequential grouping forward validation results of the liquor series.

When comparing the ordinary FV results in Table 4.12 on page 202 and the sequential grouping FV results in Table 4.17 a few reassuring conclusions are yet again made. The top two models are again the same, respectively using the 13 and 12 leading eigenvectors.

It is comforting to conclude that Algorithm 4.2 and Algorithm 4.3 produce similar results. Algorithm 4.2 is fully automated and "user-bias" is not possible, as was the case with Algorithm 4.3.

The following section will consider robust forward validation procedures.

4.4.6 Robust Forward Validation

The forward validation procedure, used in this thesis, splits an observed time series into three non-overlapping sets. Forecasting and model selection in SSA can be influenced by the possible presence of outliers in the training- or validation set. This section proposes methods for guarding against, and outlines difficulties that can arise due to, both the latter situations.

We first consider robustification of FV when an outlier(-s) is present in the training set. The training set is used to construct a trajectory matrix ($\mathbf{X}$), shift vector ($\bar{\mathbf{x}}$) and scatter matrix ($\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$). It is a well-documented fact that the classical sample mean vector ($\bar{\mathbf{x}}$) and sample variance-covariance matrix $\mathbf{S} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}'/n$ are sensitive to the presence of outliers (Huber 1981, p. 199). These measures are therefore not resistant to outliers that might possibly be lurking in the original trajectory matrix $\mathbf{X}$. To complicate matters even further, is the fact that the trajectory matrix has a Hankel matrix structure. If a single outlier is present in the training series, then this will result in an outlier being present in a number of consecutive column vectors of the trajectory matrix.

SSA uses SVD of the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$. The eigenvectors are therefore also sensitive to the presence of outliers through this dependence. A good illustrative example (in $\mathbb{R}^2$) of the effect of a single outlier on the classical sample mean and consequent alignment of principal components can be found in Locantore *et al.* (1999, pp. 11-19).

A number of procedures exist in the multivariate statistical literature that address the issue of robust PCA (Devlin *et al.*, 1981; Jackson, 1991, pp. 365-371; Maronna and Yohai, 1998, pp. 589-596; Marden, 1999; Croux and Haesbroeck, 2000; Kamiya and Eguchi, 2001) and resistant SVD (Yong-Seok and Myung-Hoe, 1996). Many of these techniques employ variations of the Mahalanobis distance measure. However, these are unfortunately not directly applicable to our situation. This is due to the fact that our "observations", viz. column vectors of the trajectory matrix, are not identically distributed (see e.g. Chapter 2), which is assumed in the definition of Mahalanobis distance.

In the case of SSA, alternative outlier detection/ robustification procedures thus need to be considered. It seems that the most effective way is to deal with the outlier(s) in the data, prior to the SVD process. This is what we currently recommend. Section 4.5 considers outlier identification in the training set.

Outliers present in the validation series could also influence FV results. The influence of such outliers is passed onto the measure of in-sample forecasting accuracy. Venter (1998) has already made a suggestion to deal with this situation. His suggestion has been incorporated into Algorithm 4.2 to form Algorithm 4.4 below.

Algorithm 4.4 Forward validation using k-period-ahead forecasting and guarding against a single outlier

(a) Load an observed time series $\{x_t\}_{t=1}^{N}$ and transform if necessary.

(b) Set the number of periods to forecast ahead (k). Split the time series into a training set $\{x_t\}_{t=1}^{N_1-k+1}$, validation set $\{x_t\}_{t=N_1+1}^{N_1+N_2}$ and test set $\{x_t\}_{t=N_1+N_2+1}^{N}$.

(c) Set the maximum number of leading eigenvectors $r_{\max}$ and the maximum window length $\tau_{\max}(\leq$ integer part of $(N_1-k+2)/2)$. Loop over the window length (τ), where $k+r_{\max} \leq \tau \leq \tau_{\max}$.

(d) Loop over the number of leading eigenvectors to use, where $1 \leq r \leq r_{\max}$ and $r_{\max}$ is the maximum number of leading eigenvectors to use during FV.

(e) Construct the Cadzow-signal series using the training set, the window length in the loop at step (c) and the number of leading eigenvectors in the loop in step (d).

(f) Unfold the Cadzow-signal series into the column vectors of a $\tau \times (N_1 - k - \tau + 2)$ trajectory matrix. Construct the centred trajectory matrix $\tilde{\mathbf{X}}$ and scatter matrix $(\tilde{\mathbf{X}}\tilde{\mathbf{X}}')$.

(g) Perform SVD of the scatter matrix $(\tilde{\mathbf{X}}\tilde{\mathbf{X}}')$.

(h) Construct the matrix $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}$, using the r leading eigenvectors.

(i) Construct the projection matrix $\mathbf{P}_{\mathfrak{L}_r}$.

(j) Form the projection matrix $\mathbf{P}_{\mathfrak{L}_r^\perp} = \mathbf{I}_\tau - \mathbf{P}_{\mathfrak{L}_r}$.

(k) Partition the projection matrix in (j) as follows

$$\begin{aligned} \mathbf{P}_{\mathfrak{L}_r^\perp} &= \left[\mathbf{P}_{\mathfrak{L}_r^\perp} \left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-k} \right] \middle| \; \mathbf{P}_{\mathfrak{L}_r^\perp} \left[\mathbf{e}_{\tau-k+1} \quad \cdots \quad \mathbf{e}_\tau \right] \right] \\ &= \left[\underset{\tau\times(\tau-k)}{\mathbf{P}_1} \middle| \underset{\tau\times k}{\mathbf{P}_2} \right], \end{aligned} \tag{4.41}$$

where $\mathbf{e}_j$ is the j^{th} column vector of the $\tau \times \tau$ identity matrix $\mathbf{I}_\tau$, and k is the number of periods being forecast ahead.

(l) Calculate the k-period-ahead forecasts using

$$\begin{bmatrix} \hat{x}_{N_1-k+2} \\ \vdots \\ \hat{x}_{N_1+1} \end{bmatrix} = \begin{bmatrix} \overline{x}_{\tau-k+1} \\ \vdots \\ \overline{x}_\tau \end{bmatrix} - \left(\mathbf{P}_2'\mathbf{P}_2 \right)^{-1} \left(\mathbf{P}_2'\mathbf{P}_1 \right) \begin{bmatrix} x_{N_1-\tau+2} - \overline{x}_1 \\ \vdots \\ x_{N_1-k+1} - \overline{x}_{\tau-k} \end{bmatrix}, \tag{4.42}$$

where

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{P}_{\mathfrak{L}_r^\perp} \left[\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{\tau-k} \right] \\ \mathbf{P}_2 &= \mathbf{P}_{\mathfrak{L}_r^\perp} \left[\mathbf{e}_{\tau-k+1} \quad \cdots \quad \mathbf{e}_\tau \right] . \end{aligned}$$

(m) If a transformation was applied to the series in step (a) then a back-transformation must be used in this step. Calculate and order the k forecast errors from small to large and ***drop the largest forecast error***. Calculate the mean squared prediction error $\frac{1}{k-1}\sum_{i=1}^{k-1}\left(x_{(N_1-k+i+1)} - \hat{x}_{(N_1-k+i+1)} \right)^2$. Increase the training set length to $N_1 = N_1 + 1$ and return to step (e). The process is continued until the whole validation set has been forecast k-periods-ahead.

(n) Form elements of the following matrix

$$\underset{(\tau_{\max}-k-r_{\max}+1)\times r_{\max}}{\mathbf{MSE}} = \begin{bmatrix} mse_{k+r_{\max},1} & mse_{k+r_{\max},2} & \cdots & mse_{k+r_{\max},r_{\max}} \\ mse_{k+r_{\max}+1,1} & mse_{k+r_{\max}+1,2} & \cdots & mse_{k+r_{\max}+1,r_{\max}} \\ \vdots & \vdots & \vdots & \vdots \\ mse_{\tau_{\max},1} & mse_{\tau_{\max},2} & \cdots & mse_{\tau_{\max},r_{\max}} \end{bmatrix}, \quad (4.43)$$

where

$$mse_{i,j} = \frac{1}{(k-1)N_2} \sum_{t=N_1+1}^{N_1+N_2} \sum_{i=1}^{k-1} \left(x_{(t-k+i)} - \hat{x}_{(t-k+i)} \right)^2 \quad (4.44)$$

for $k + r_{\max} \leq i \leq \tau_{\max} (\leq$ integer part of $(N_1 - k + 2)/2)$ and $1 \leq j \leq r_{\max}$.

(o) Calculate the following measure

$$FV_j = \underbrace{\left[\frac{1}{\tau_{\max} - r_{\max} - k + 1} \sum_{i=k+r_{\max}}^{\tau_{\max}} mse_{i,j} \right]^2}_{\text{mean(MSE)}^2}$$
$$+ \underbrace{\frac{1}{\tau_{\max} - r_{\max} - k} \sum_{i=k+r_{\max}}^{\tau_{\max}} \left[mse_{i,j} - \left\{ \frac{1}{\tau_{\max} - r_{\max} - k + 1} \sum_{i=k+r_{\max}}^{\tau_{\max}} mse_{i,j} \right\} \right]^2}_{\text{var(MSE)}}$$

(4.45)

for $1 \leq j \leq r_{\max}$.

Remark 4.4

Robustification in this proposal consists of dropping the largest prediction error. One can, alternatively, drop (say) 5% of the largest prediction errors, to guard against the presence of more than one outlier. Although this is straightforward to implement, we found that dropping the largest error suffices for our purposes.

4.5 Outlier Identification in SSA

The following section will be used to propose a method for identifying a single additive outlier in an observed time series, in the context of SSA. It is assumed throughout this section that the column vectors of the trajectory matrix are multivariate normally distributed. Cadzow's (1988) composite property mapping algorithm is employed to smooth out noise from an observed series, yielding a reconstructed signal series that conforms to the SSA assumptions. A residual series is then formed by subtracting the Cadzow-signal series from the observed time series. The residual series is then unfolded into a trajectory matrix. This places the identification of an outlier into a multivariate context. It was shown in Chapter 2, under a normal assumption, that a column vector of the residual trajectory matrix has a multivariate normal distribution of the form

$$\mathbf{e}_j \sim N_\tau(\mathbf{0},\ \Sigma). \tag{4.46}$$

The column vectors of the residual trajectory matrix are therefore identically distributed. This makes outlier identification using the Mahalanobis distance possible. According to Kosinski (1999, p. 146)

"*Multivariate outliers can be identified as points with large Mahalanobis distances based on robust estimates of population scatter and location.*"

It is also important to take note of the fact that an outlier in a time series translates to a number of outliers in the SSA context. This is because the Hankel structure of the residual trajectory matrix will cause a single outlier to be present in a number of consecutive column vectors. If an outlier is, for example, present at observation x_l, then it will be present in column vectors $\mathbf{e}_{j_1},\ldots,\mathbf{e}_{j_2}$, where

$$\begin{aligned} j_1 &= \max(1, l-\tau+1) \\ j_2 &= \min(l, n). \end{aligned} \tag{4.47}$$

Hence, a single outlier will translate to anomalies in the Mahalanobis distances of column vectors $\mathbf{e}_{j_1},\ldots,\mathbf{e}_{j_2}$. This is how a single outlier will be searched for.

The study of multivariate outliers dates back to work done by Wilks (1963). It is well documented that the Mahalanobis distance measure fails to detect outliers due to the masking effect (Becker and Gather, 1999; Rousseeuw and Van Zomeren, 1990). This effect occurs when the Mahalanobis distance measure uses the classical sample mean vector $(\overline{\mathbf{x}})$ and sample covariance matrix $(\mathbf{S})$. Outliers influence both the latter measures and by doing so mask their activities. The result is that the Mahalanobis distance measure is affected and an outlier may pass a test undetected.

Various other methods have also been devised to detect multivariate outliers. According to Becker and Gather (1999)

"*Various concepts for outlier identification in multivariate samples exist in the literature. Among them are methods of heuristic nature (e.g. Atkinson and Mulira 1993; Bacon-Shone and Fung 1987; Barnett and Lewis 1994, p. 307; Bhandary 1992) and those of a consecutive testing type (Barnett and Lewis 1994, chap. 7.3; Caroni and Prescott 1992; Hara 1988; Hawkins 1980, chap. 9; Wilks 1963).*"

In this section we make use of a fairly recent method proposed by Rousseeuw and Van Driessen (1999). The method can be classified as an S-estimator, as it attempts to find a certain subset of multivariate observations that minimizes the determinant of the covariance matrix. The technique is extremely computationally efficient and can handle many multivariate observations with ease. This is a situation that can definitely occur in SSA.

Algorithm 4.5 Outlier Identification in SSA by using FAST-MCD

(a) Load an observed time series $\{x_t\}_{t=1}^{N}$ and transform if necessary.

(b) Reconstruct the Cadzow-signal series for the series in step (a) using a window length τ and r eigenvectors.

(c) Subtract the Cadzow-signal (back transformed) from the series in (a) to form the residual series $\left\{e_t\right\}_{t=1}^{N}$.

(d) Unfold the residual series into the $\tau \times n(= N - \tau + 1)$ residual trajectory matrix $\mathbf{E}$.

(e) Use the FAST-MCD routine of Rousseeuw and Van Driessen (1999) to calculate robust measures of location $\left(\bar{e}^*\right)$ and spread $\left(\mathbf{S}^*\right)$, based on the matrix in (d).

(f) The robust measures of location $\left(\bar{e}^*\right)$ and spread $\left(\mathbf{S}^*\right)$, in step (e), are then used to calculate the Mahalanobis distance for each column vector, viz.

$$d_i^* = \sqrt{\left(e_i - \bar{e}^*\right)' \mathbf{S}^{*-1} \left(e_i - \bar{e}^*\right)} \tag{4.48}$$

for $i = 1, \ldots, n(= N - \tau + 1)$.

(g) Define a cut-off point as $\sqrt{\chi^2_{\tau;1-\alpha}}$. Column vectors of the residual trajectory matrix, in step (d), with Mahalanobis distances $\left(d_i^*\right)$ much larger than the cut-off are identified as containing an outlier.

(h) If column vector $\mathbf{e}_{j_1}$ is identified in step (g), then the position of the outlier is given by

$$l = j_1 + \tau - 1. \tag{4.49}$$

To verify the outlier's position, note that the index of the last column vector to contain an outlier, and therefore a large Mahalanabis distance $\left(d_i^*\right)$, must be

$$j_2 = \min(l, n).$$

Remark 4.5

(a) The reconstruction of a Cadzow-signal series assumes that the number of leading eigenvectors to use, are known. If this is not the case, an artificial phase space portrait can be constructed for the purpose of identifying structure.

(b) The Cadzow-signal series has been found in this study to be very robust against outliers. The reconstruction of this series is iterative by nature and the effect of outliers is down weighted in this scheme due to the Hankelization operation that is used. There should, therefore, not be great concern that structure will be compromised in this regard.

(c) Section 2.6.1 in Chapter 2 can be consulted for distributional assumptions regarding the residual trajectory matrix.

Example 4.14 Identifying a single outlier in the case of the airline series

A single additive outlier was placed in the airline series at observation $x_{123} = 606$. The true observation was $x_{123} = 406$. The airline series was then log-transformed. A window length of $\tau = 36$ and the 13 leading eigenvectors were used to reconstruct the Cadzow-signal series. The Cadzow-signal was then back transformed and subtracted from the airline series containing the outlier, forming a residual series. The residual series was then used to construct the $\tau \times n$ residual trajectory matrix $\mathbf{E}$. The residual trajectory matrix had 109 column vectors and was then supplied to the FAST-MCD routine of Rousseeuw and Van Driessen (1999) for outlier identification. The results are summarised in Table 4.18 below.

Table 4.18 Outlier identification using FAST-MCD.

Column Vector (j)	Classical Mahalanobis Distance	Robust Mahalanobis Distance	Column Vector (j)	Classical Mahalanobis Distance	Robust Mahalanobis Distance
85	5.15286	6.421830	**98**	9.76616	**38.80945**
86	5.22036	9.806900	**99**	9.80033	**38.56285**
87	5.12127	10.61205	**100**	9.8088	**44.61752**
88	9.46309	**37.91700**	**101**	9.80679	**48.88773**
89	9.49954	**38.18633**	**102**	9.80936	**49.79763**
90	9.50145	**37.49795**	**103**	9.80823	**45.83956**
91	9.46009	**35.66818**	**104**	9.80928	**46.03890**
92	9.4643	**32.63775**	**105**	9.84555	**45.51619**
93	9.50761	**36.16598**	**106**	9.86962	**46.35750**
94	9.51000	**39.71524**	**107**	9.89691	**47.24311**
95	9.54531	**36.73054**	**108**	9.8812	**43.76550**
96	9.55073	**34.99734**	**109**	9.91135	**45.20758**
97	9.53579	**33.03999**			

The FAST-MCD program also returned the square root of the chi-square critical value, viz.

$$\sqrt{\chi^2_{36;0.975}} = 7.38\,. \tag{4.50}$$

This was used as cut-off value in the FAST-MCD routine. The column vectors in the residual trajectory matrix that had Mahalanobis distances greater than this value, were not used in creating the robust measures of location and spread. The latter were then used to calculate the robust Mahalanobis distance measure. According to Rousseeuw and Van Driessen (1999), column vectors with robust Mahalanobis distance measures that are "*substantially larger than the cut-off should be considered outliers*".

From perusal of Table 4.18, above, it is clear that the column vectors starting at index 88 and ending at 109 have robust Mahalanobis distances that are substantially larger than the cut-off value in (4.50). Step (h) in Algorithm 4.5 states that the position of an outlier is calculated by identifying the first column vector with large robust Mahalanobis distance as follows

$$l = 88 + 36 - 1 = 123\,.$$

This implies that an outlier is present at position $t = 123$. The outlier is, however, present in a number of consecutive column vectors. The last column vector to contain the outlier is determined by

$$j_2 = \min(l,n) = \min(123,109) = 109.$$

It is clear from the results that the last column vector with large robust Mahalanobis distance is indeed column vector 109.

Remark 4.6

(a) Once an outlier is identified there exists a number of ways that one can deal with the situation. One possible way is to leave the observation out and to replace it with the linearly interpolated result of the observations on either side of it.

(b) It is also important to note that the method, proposed in this section, can only deal with a single additive outlier. If two or more outliers are present in a time series, then a number of complications can arise. The most serious one being when outliers are very close together in the time series it might be impossible to identify them, using the column vectors with large Mahalanobis distances. The exact location of the outliers might be impossible to determine as there will be multiple outliers in consecutive column vectors.

The above concludes the identification of an outlier in the SSA context. We will now shift our attention to Monte Carlo simulations. The following section will therefore illustrate the effectiveness of the forward validation routine, proposed in Algorithm 4.2, through the use of a small scale Monte Carlo simulation study.

4.6 Monte Carlo Simulations

A small scale Monte Carlo simulation study was conducted as part of this study. A few noise-contaminated series were simulated using a signal from the broad class of functions proposed by Buchstaber (1994), viz.

$$f(t) = \sum_{k=1}^{K} p_k(t) \exp(\lambda_k t) \sin(\varpi_k t + \varphi_k) \tag{4.51}$$

where $p_k(t)$ are polynomials.

It is evident from (4.51) that a rather unmanageable number of Monte Carlo simulations can result from this class of functions. It is natural that concern about the vastness of such an exercise can be raised in that one will be forced to a sparse factorial design.

We confined ourselves to three models, viz. a linear trend model, a model with cyclical variation and a model with linear trend and cyclical variation.

The out-of-sample forecasting accuracy performances of FV selected SSA models were compared with results obtained from the Forecast Pro XE (Version 3) software package. Batch mode operation with expert selection was used in this package. In this mode the Forecast Pro XE package searches for the optimum model from a broad class of alternative models, which includes exponential smoothing models (simple, Holt, Winters) and the broad class of Box-Jenkins models. According to the Forecast Pro XE user's manual

"*The expert system uses a production rule logic that depends upon the statistical properties of data. If the system needs to decide between Box-Jenkins and exponential smoothing, it makes its selection via out-of-sample comparison of accuracy (rolling simulation).*"

The package is quoted in leading forecasting journals as being one of the best in its class and has won many forecasting competitions.

Example 4.15 Monte Carlo simulations for a linear trend model

In this example 100 Monte Carlo repetitions, $\left\{x_t^{(1)}\right\}_{t=1}^{100},\ldots,\left\{x_t^{(100)}\right\}_{t=1}^{100}$ were simulated from the following model

$$x_t = \begin{cases} 10 + 2t + \varepsilon_t & for \quad t = 1,\ldots,90 \\ 10 + 2t & for \quad t = 91,\ldots,100, \end{cases} \tag{4.52}$$

where

$$\varepsilon_t \sim N\left(0,4^2\right).$$

It is clear that the model in (4.52) constitutes an additive model with linear trend and white noise in the first 90 observations. All the Monte Carlo repetitions, i.e. time series of length $N = 100$, consisted of a noise contaminated signal $\left\{x_t\right\}_{t=1}^{90}$ and noise-free signal $\left\{x_t\right\}_{t=91}^{100}$.

Each Monte Carlo series $\left\{x_t\right\}_{t=1}^{100}$ was split into three sets, a training set $\left\{x_t\right\}_{t=1}^{80}$, validation set $\left\{x_t\right\}_{t=81}^{90}$ and test set $\left\{x_t\right\}_{t=91}^{100}$. The forward validation algorithm, proposed in Algorithm 4.2, was used in all the cases to perform model selection. The number of periods to forecast ahead was set at $k = 10$. The maximum window length to use was set at $\tau_{\max} = 36$. The maximum number of leading eigenvectors used during FV was set at $r_{\max} = 6$.

The simulated model in (4.52) is a rank 1 model. This implies that the leading eigenvector should describe the model adequately in the case of a noise-contaminated series.

The forward validation model selection results are summarised in Table 4.19 below. The table contains the number of leading eigenvectors in a model and the number of Monte Carlo repetitions (series) that were placed into each of these categories during FV.

Table 4.19 Model selection results of the Monte Carlo simulations.

Leading Eigenvectors (r)	Number of Models
1	68
2	16
3	7
4	2
5	4
6	3
	100

It is clear from the above table that the majority of models (68%) were correctly classified. Two leading eigenvectors can also describe a linear trend and should also be taken as correct in the above case. This would yield a "correct classification rate" equal to 84%.

The models selected using FV were then used to test out-of-sample accuracy. Out-of-sample accuracy was calculated over the noise-free test set $\{x_t\}_{t=91}^{100}$ for each Monte Carlo repetition. Rolling recurrent one-period-ahead forecasting was used for this purpose. Centring of the trajectory matrix was used and the original time series was used to perform forecasting. Forecast Pro was also used to produce out-of-sample forecasts for the test set. The hold-out-sample was therefore set equal to the last 10 observations of each Monte Carlo repetition. The batch forecasting option of the Forecast Pro XE software package was used with expert model selection. The mean of the MAD, MSE and MAPE measures of forecasting accuracy were calculated for the SSA and Forecast Pro XE results as follows

$$\begin{aligned}\overline{MAD} &= \frac{1}{(100)(10)}\sum_{i=1}^{100}\sum_{t=91}^{100}\left|x_t^{(i)} - \hat{x}_t^{(i)}\right| \\ \overline{MSE} &= \frac{1}{(100)(10)}\sum_{i=1}^{100}\sum_{t=91}^{100}\left(x_t^{(i)} - \hat{x}_t^{(i)}\right)^2 \\ \overline{MAPE} &= \frac{1}{(100)(10)}\sum_{i=1}^{100}\sum_{t=91}^{100}\left|x_t^{(i)} - \hat{x}_t^{(i)}\right| / x_t^{(i)} \times 100\%.\end{aligned} \tag{4.53}$$

Table 4.20 Out-of-sample measures of forecasting accuracy for Monte Carlo results.

Measure / Method	SSA (FV selected model)	Forecast Pro XE
$\overline{MAD}$	0.5763	1.0653
$\sqrt{\overline{MSE}}$	0.7482	2.6352
$\overline{MAPE}$ (%)	0.2880	0.5264

It is clear from Table 4.20, above, that SSA outperformed the results obtained from the Forecast Pro XE package.

Example 4.16 Monte Carlo simulations for a cyclical model

In this example 100 Monte Carlo repetitions, $\left\{x_t^{(1)}\right\}_{t=1}^{100},\ldots,\left\{x_t^{(100)}\right\}_{t=1}^{100}$ were simulated from the following model

$$x_t = \begin{cases} \sin\left(2\pi t/12+10\right)+\varepsilon_t & for \quad t=1,\ldots,90 \\ \sin\left(2\pi t/12+10\right) & for \quad t=91,\ldots,100, \end{cases} \tag{4.54}$$

where

$$\varepsilon_t \sim N\left(0,4^2\right).$$

It is clear that the model in (4.54) constitutes a model with cyclical variation with period 12 and additive white noise in the first 90 observations. All the Monte Carlo repetitions, i.e. time series of length $N=100$, consisted of a noise contaminated signal $\left\{x_t\right\}_{t=1}^{90}$ and noise-free signal $\left\{x_t\right\}_{t=91}^{100}$.

Each Monte Carlo series $\left\{x_t\right\}_{t=1}^{100}$ was split into three sets, a training set $\left\{x_t\right\}_{t=1}^{80}$, validation set $\left\{x_t\right\}_{t=81}^{90}$ and test set $\left\{x_t\right\}_{t=91}^{100}$. The forward validation algorithm, proposed in Algorithm 4.2, was used in all the cases. The number of periods to forecast ahead was set at $k=10$. The maximum window length to use was set at $\tau_{\max}=36$. The maximum number of leading eigenvectors used during FV was set at $r_{\max}=6$.

The simulated model in (4.54) is a rank 2 model. This implies that the two leading eigenvectors should describe the model adequately in the case of a noise-contaminated series.

The forward validation model selection results are summarised in Table 4.21 below. The table contains the number of leading eigenvectors in a model and the number of Monte Carlo repetitions (series) that were placed into each of these categories during FV.

Table 4.21 Model selection results of the Monte Carlo simulations.

Leading Eigenvectors (r)	Number of Models
1	0
2	77
3	5
4	9
5	4
6	5
	100

It is clear from the above table that the majority of models (77%) were correctly classified.

The models selected using FV were then used to test out-of-sample accuracy. Out-of-sample accuracy was calculated over the noise-free test set $\{x_t\}_{t=91}^{100}$ for each Monte Carlo repetition. Rolling recurrent one-period-ahead forecasting was used for this purpose. Centring of the trajectory matrix was used and the original time series was used to perform forecasting. Forecast Pro was also used to produce out-of-sample forecast for the test set. The hold-out-sample was therefore set equal to the last 10 observations of each Monte Carlo repetition. The batch forecasting option of the Forecast Pro XE software package was used with expert model selection. The mean of the MAD, MSE and MAPE measures of forecasting accuracy were calculated for the SSA and Forecast Pro XE results as follows

$$\begin{aligned}\overline{MAD} &= \frac{1}{(100)(10)}\sum_{i=1}^{100}\sum_{t=91}^{100}\left|x_t^{(i)} - \hat{x}_t^{(i)}\right| \\ \overline{MSE} &= \frac{1}{(100)(10)}\sum_{i=1}^{100}\sum_{t=91}^{100}\left(x_t^{(i)} - \hat{x}_t^{(i)}\right)^2 \\ \overline{MAPE} &= \frac{1}{(100)(10)}\sum_{i=1}^{100}\sum_{t=91}^{100}\left|x_t^{(i)} - \hat{x}_t^{(i)}\right| / x_t^{(i)} \times 100\%.\end{aligned} \tag{4.55}$$

Table 4.22 Out-of-sample measures of forecasting accuracy for Monte Carlo results.

Measure / Method	SSA (FV selected model)	Forecast Pro XE
$\overline{MAD}$	1.1274	1.2697
$\sqrt{\overline{MSE}}$	1.4426	1.6226
$\overline{MAPE}$ (%)	36.6740	42.6330

It is clear from Table 4.22, above, that SSA slightly outperformed the results obtained from the Forecast Pro XE package. Note that the large MAPE figures in Table 4.22 are due to the scale of the noise variance relative to the signal series, which oscillates between the values -1 and 1.

Example 4.17 Monte Carlo simulations for a model with linear trend and cyclical variation

In this example 100 Monte Carlo repetitions, $\left\{x_t^{(1)}\right\}_{t=1}^{100},\ldots,\left\{x_t^{(100)}\right\}_{t=1}^{100}$ were simulated from the following model

$$x_t = \begin{cases} (50+2t)+\sin\left(2\pi t/12+55\right)+\varepsilon_t & for \quad t=1,\ldots,90 \\ (50+2t)+\sin\left(2\pi t/12+55\right) & for \quad t=91,\ldots,100, \end{cases} \tag{4.56}$$

where

$$\varepsilon_t \sim N\left(0,15^2\right).$$

It is clear that the model in (4.56) constitutes a linear trend model with cyclical variation with period 12 and additive white noise in the first 90 observations. All the Monte Carlo repetitions, i.e. time series of length $N=100$, consisted of a noise contaminated signal $\left\{x_t\right\}_{t=1}^{90}$ and noise-free signal $\left\{x_t\right\}_{t=91}^{100}$.

Each Monte Carlo series $\left\{x_t\right\}_{t=1}^{100}$ was split into three sets, a training set $\left\{x_t\right\}_{t=1}^{80}$, validation set $\left\{x_t\right\}_{t=81}^{90}$ and test set $\left\{x_t\right\}_{t=91}^{100}$. The forward validation algorithm, proposed in Algorithm 4.2, was used in all the cases. The number of periods to forecast ahead was set at $k = 10$. The maximum window length to use was set at $\tau_{\max} = 36$. The maximum number of leading eigenvectors used during FV was set at $r_{\max} = 6$.

The simulated model in (4.56) is a rank 3 model. This implies that the three leading eigenvectors should describe the model adequately in the case of a noise-contaminated series.

The forward validation model selection results are summarised in Table 4.23 below. The table contains the number of leading eigenvectors in a model and the number of Monte Carlo repetitions (series) that were placed into each of these categories during FV.

Table 4.23 Model selection results of the Monte Carlo simulations.

Leading Eigenvectors (r)	Number of Models
1	0
2	0
3	73
4	9
5	16
6	2
	100

It is clear from the above table that the majority of models (73%) were correctly classified. It is interesting to note that 16% of the models were classified as models using $r = 5$ leading eigenvectors. The model in (4.56) would usually require only $r = 3$ leading eigenvectors. It is clear that the model selection used an extra sinusoidal when using $r = 5$ leading eigenvectors. Recall that a sinusoidal requires an additional 2 leading eigenvectors. An explanation for this situation could be viewed as an additional harmonic that was introduced to model the simulated series. The model selection result of $r = 5$ is therefore not entirely parsimonious but is also not an excessive over parameterisation of a simulated series, given the non-parametric nature of SSA.

The models selected using FV were then the used to test out-of-sample accuracy. Out-of-sample accuracy was calculated over the noise-free test set $\{x_t\}_{t=91}^{100}$ for each Monte Carlo repetition. Rolling recurrent one-period-ahead forecasting was used for this purpose. Centring of the trajectory matrix was used and the original time series was used to perform forecasting. Forecast Pro was also used to produce out-of-sample forecast for the test set. The hold-out-sample was therefore set equal to the last 10 observations of each Monte Carlo repetition. The batch forecasting option of the Forecast Pro XE software package was used with expert model selection. The mean of the MAD, MSE and MAPE measures of forecasting accuracy were calculated for the SSA and Forecast Pro XE results as follows

$$\begin{aligned}\overline{MAD} &= \frac{1}{(100)(10)}\sum_{i=1}^{100}\sum_{t=91}^{100}\left|x_t^{(i)} - \hat{x}_t^{(i)}\right| \\ \overline{MSE} &= \frac{1}{(100)(10)}\sum_{i=1}^{100}\sum_{t=91}^{100}\left(x_t^{(i)} - \hat{x}_t^{(i)}\right)^2 \\ \overline{MAPE} &= \frac{1}{(100)(10)}\sum_{i=1}^{100}\sum_{t=91}^{100}\left|x_t^{(i)} - \hat{x}_t^{(i)}\right| / x_t^{(i)} \times 100\%.\end{aligned} \tag{4.57}$$

Table 4.24 Out-of-sample measures of forecasting accuracy for Monte Carlo results.

Measure / Method	SSA (FV selected model)	Forecast Pro XE
$\overline{MAD}$	5.8155	9.7662
$\sqrt{\overline{MSE}}$	7.3204	12.9172
$\overline{MAPE}$ (%)	2.5282	4.2909

It is clear from Table 4.24, above, that SSA outperformed the results obtained from the Forecast Pro XE package.

4.7 Other Examples

This example performed forward validation and applied Algorithm 4.2 to 3003 time series. The time series formed part of the M3-IJF competition. This competition applied various time series forecasting methods to the time series and the results were published in *The International Journal of Forecasting*. Makridakis *et al.* (1982) can be consulted for a similar forecasting competition in which 1001 time series were used.

The time series, used in this example, can be downloaded from the following internet website www.maths.monash.edu.au/~hyndman/forecasting/ , which is maintained by Prof Rob Hyndman.

These time series varied in length from as short as 14 observations to a maximum length of 126 observations. Each time series was split into three non-overlapping parts during FV, viz. a training series a validation series and test series.

The length of each series was set as follows

$$N_1 = \begin{cases} \min\left(\text{integer part of } (0.1 \times N),\ 12\right) & \textit{if} \quad \text{integer part of } (0.1 \times N) > 3 \\ 3 & \textit{if} \quad \text{integer part of } (0.1 \times N) \le 3 \end{cases}$$
$$N_2 = N_1$$
$$N_3 = N - N_1 - N_2, \tag{4.58}$$

where

$N =$ length of time series

$N_1 =$ length of training series

$N_2 =$ length of validation series

$N_3 =$ length of test series.

To set the maximum number of leading eigenvectors, the following window length was used

$$\tau_{\max} = \min\left(\text{integer part of } \left(N_1 + 1\right)/2,\ 70\right). \tag{4.59}$$

The maximum number of leading eigenvectors to use was then set at

$$r_{\max} = \begin{cases} \min\left(\tau_{\max} - 7,\ 13\right) & if \quad \tau_{\max} - 7 > 0 \\ \min\left(\tau_{\max} - 2,\ 13\right) & if \quad \tau_{\max} - 7 \leq 0 \end{cases}. \tag{4.60}$$

The number of periods to forecast ahead was then set as follows

$$k = \begin{cases} \min\left(N_1 - \tau_{\max} - 1,\ 12\right) & if \quad N_1 - \tau_{\max} \geq 6 \\ 1 & if \quad N_1 - \tau_{\max} < 6 \end{cases}. \tag{4.61}$$

The maximum window length eventually used, was set at

$$\tau_{\max} = \text{integer part of } \left(N_1 - k + 2\right)/2. \tag{4.62}$$

Each forward validation selected model was used to generate rolling one-period-ahead forecasts over the test series. The whole process used 7 days of processor time (800mHz). Table 4.25 reports the models that were selected. Table 4.26 reports the out-of-sample MAPE measure of forecasting accuracy. These results are cumulative figures and ordered in ascending order in multiples of 100 models.

A few interesting facts can be deduced from perusal of Table 4.25, below. The majority of FV selected models used only one leading eigenvector, in the case of series of length less than 100 observations. These series are very short and forward validation model selection was only able to use rolling one-period-ahead forecasting for many of them. It is recommended that series with length of at least 100 observations be used during application of the FV algorithms proposed in this study. It is also recommended that $k \geq 6$ period-ahead-forecasting be used during forward validation model selection.

It was mentioned in the previous paragraph that the majority $(1949 \text{ of } 3003)$ of FV selected models used only the leading eigenvector. Models of this nature can be considered as trend models that exhibit a linear or exponential trend. The second and third most popular model choices were the two $(297 \text{ of } 3003)$ and three $(259 \text{ of } 3003)$ leading eigenvectors. It is clear that these model selection choices resulted from series of length 100 observations and longer.

Judging by the results reported in Table 4.26, below, the out-of-sample forecasting performance is reasonable. The top 1000, 2000 and 3000 models respectively had out-of-sample MAPE figures of 1.92%, 4.47% and 12.91%. It would seem that a large number of "adequate" models produced good results. Models that produced inadequate results performed dismally to produce an overall MAPE figure of 14.05% for the entire ensemble of time series. This can only be ascribed to the fact that there are time series present which do not have structure that the SSA technique can handle.

Table 4.25 Forward validation results of 3003 time series.

	Number of Leading Eigenvectors (r)													
N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**Grand Total**
14	121	31	0	0	0	0	0	0	0	0	0	0	0	152
15	15	13	12	0	0	0	0	0	0	0	0	0	0	40
16	28	21	21	0	0	0	0	0	0	0	0	0	0	70
17	61	11	20	11	0	0	0	0	0	0	0	0	0	103
18	2	1	2	0	0	0	0	0	0	0	0	0	0	5
19	45	24	27	18	15	0	0	0	0	0	0	0	0	129
20	2	0	1	1	1	0	0	0	0	0	0	0	0	5
21	2	0	0	0	0	0	0	0	0	0	0	0	0	2
22	3	0	0	0	0	0	0	0	0	0	0	0	0	3
24	4	0	0	0	0	0	0	0	0	0	0	0	0	4
25	3	0	0	0	0	0	0	0	0	0	0	0	0	3
26	18	0	0	0	0	0	0	0	0	0	0	0	0	18
27	15	0	0	0	0	0	0	0	0	0	0	0	0	15
28	2	0	0	0	0	0	0	0	0	0	0	0	0	2
29	5	0	0	0	0	0	0	0	0	0	0	0	0	5
30	13	0	0	0	0	0	0	0	0	0	0	0	0	13
31	22	0	0	0	0	0	0	0	0	0	0	0	0	22
32	5	0	0	0	0	0	0	0	0	0	0	0	0	5
34	5	0	0	0	0	0	0	0	0	0	0	0	0	5
35	34	0	0	0	0	0	0	0	0	0	0	0	0	34
36	173	0	0	0	0	0	0	0	0	0	0	0	0	173
37	26	0	0	0	0	0	0	0	0	0	0	0	0	26
38	27	0	0	0	0	0	0	0	0	0	0	0	0	27
39	7	0	0	0	0	0	0	0	0	0	0	0	0	7
40	33	0	0	0	0	0	0	0	0	0	0	0	0	33
41	88	0	0	0	0	0	0	0	0	0	0	0	0	88
42	5	0	0	0	0	0	0	0	0	0	0	0	0	5
43	9	0	0	0	0	0	0	0	0	0	0	0	0	9
44	249	0	0	0	0	0	0	0	0	0	0	0	0	249
45	33	0	0	0	0	0	0	0	0	0	0	0	0	33
46	2	0	0	0	0	0	0	0	0	0	0	0	0	2
47	1	0	0	0	0	0	0	0	0	0	0	0	0	1
48	7	0	0	0	0	0	0	0	0	0	0	0	0	7
49	1	0	0	0	0	0	0	0	0	0	0	0	0	1
50	20	0	0	0	0	0	0	0	0	0	0	0	0	20

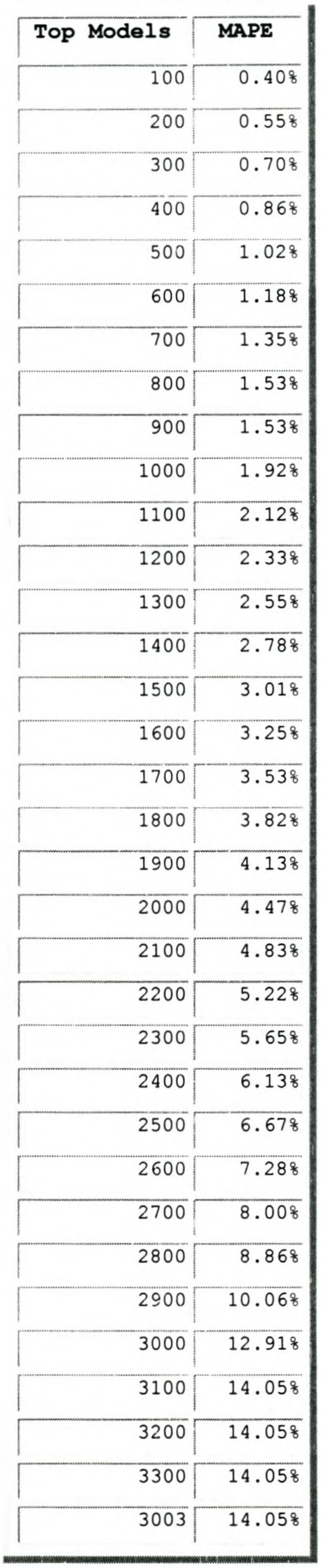
Table 4.25 (continued) Forward validation results of 3003 time series.

	Number of Leading Eigenvectors (r)													
N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**Grand Total**
51	275	0	0	0	0	0	0	0	0	0	0	0	0	275
52	3	0	0	0	0	0	0	0	0	0	0	0	0	3
53	35	0	0	0	0	0	0	0	0	0	0	0	0	35
54	18	0	0	0	0	0	0	0	0	0	0	0	0	18
55	4	0	0	0	0	0	0	0	0	0	0	0	0	4
56	20	0	0	0	0	0	0	0	0	0	0	0	0	20
58	7	0	0	0	0	0	0	0	0	0	0	0	0	7
59	1	0	0	0	0	0	0	0	0	0	0	0	0	1
60	9	0	0	0	0	0	0	0	0	0	0	0	0	9
61	6	0	0	0	0	0	0	0	0	0	0	0	0	6
62	9	0	0	0	0	0	0	0	0	0	0	0	0	9
63	144	0	0	0	0	0	0	0	0	0	0	0	0	144
64	19	0	0	0	0	0	0	0	0	0	0	0	0	19
65	3	0	0	0	0	0	0	0	0	0	0	0	0	3
66	2	0	0	0	0	0	0	0	0	0	0	0	0	2
68	10	0	0	0	0	0	0	0	0	0	0	0	0	10
69	1	0	0	0	0	0	0	0	0	0	0	0	0	1
70	4	0	0	0	0	0	0	0	0	0	0	0	0	4
71	16	0	0	0	0	0	0	0	0	0	0	0	0	16
76	0	0	0	1	0	0	0	0	0	0	0	0	0	1
78	2	13	1	3	1	1	0	1	2	0	0	0	0	24
80	1	1	2	0	0	0	1	0	0	0	0	1	0	6
81	0	1	0	0	0	0	0	0	0	0	0	0	0	1
82	1	0	0	0	0	0	0	0	1	0	0	0	0	2
83	1	0	0	0	0	0	0	0	0	0	0	0	0	1
84	1	0	0	0	0	0	0	0	0	0	0	0	0	1
86	1	0	1	0	0	0	0	0	0	0	0	0	0	2
89	0	1	0	0	0	0	0	0	1	0	0	0	0	2
90	1	0	1	0	2	0	3	0	0	0	0	1	0	8
92	1	5	1	1	0	0	1	0	0	0	0	0	0	9
94	1	0	0	0	0	0	0	0	0	0	0	0	0	1
95	3	3	1	0	2	0	0	0	0	0	0	0	0	9
96	2	3	2	1	1	2	1	0	0	1	0	0	0	13

Table 4.25 (continued) Forward validation results of 3003 time series.

	Number of Leading Eigenvectors (r)													
N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**Grand Total**
97	1	0	0	0	0	0	0	0	0	0	0	0	0	1
102	7	0	2	1	0	0	0	0	0	0	0	0	0	10
103	4	0	0	0	0	1	1	0	1	0	2	0	0	9
104	1	1	1	0	0	0	0	0	0	0	0	0	0	3
108	43	54	40	18	7	6	8	3	2	1	5	5	5	197
110	1	0	0	0	0	0	0	0	0	0	0	0	0	1
111	1	0	0	0	0	0	0	0	0	0	0	0	0	1
114	1	1	3	0	1	2	3	1	1	0	0	5	1	19
115	19	15	16	4	7	8	7	4	1	4	2	2	4	93
116	105	34	68	10	42	11	13	12	10	5	11	18	9	348
117	10	10	12	7	4	5	3	1	4	3	3	2	0	64
118	0	0	1	0	0	0	0	0	1	0	0	0	0	2
119	0	0	0	0	0	0	1	0	0	0	0	0	0	1
120	3	1	0	0	0	0	0	0	0	0	1	0	0	5
121	2	1	0	0	1	1	0	1	0	0	2	1	0	9
122	1	0	0	0	1	1	1	0	1	0	1	0	0	6
123	3	3	1	0	1	0	2	0	2	0	0	0	0	12
124	1	3	0	0	1	0	1	0	0	0	1	0	0	7
125	2	2	0	1	0	0	0	2	0	0	0	1	0	8
126	57	44	23	22	15	11	8	11	4	6	8	9	7	225
Grand Total	1949	297	259	99	102	49	54	36	31	20	36	45	26	3003

Table 4.26 Forward validation results of 3003 time series.



Top Models	MAPE
100	0.40%
200	0.55%
300	0.70%
400	0.86%
500	1.02%
600	1.18%
700	1.35%
800	1.53%
900	1.53%
1000	1.92%
1100	2.12%
1200	2.33%
1300	2.55%
1400	2.78%
1500	3.01%
1600	3.25%
1700	3.53%
1800	3.82%
1900	4.13%
2000	4.47%
2100	4.83%
2200	5.22%
2300	5.65%
2400	6.13%
2500	6.67%
2600	7.28%
2700	8.00%
2800	8.86%
2900	10.06%
3000	12.91%
3100	14.05%
3200	14.05%
3300	14.05%
3003	14.05%

4.8 Conclusions and Recommendations

An in-depth study of model selection in SSA was conducted in this chapter.

Selection of the window length (τ) was considered in Section 4.3. It was recommended in that section that the window length be set equal to an integer multiple of the highest periodicity, present in a time series. The choice of the window length is also dictated by the number of periods being forecast ahead, when using the joint-horizon k-period-ahead forecasting algorithm, proposed in Chapter 3.

Section 4.4 considered the selection of a number of leading eigenvectors. Use of a scree-diagram, the cumulative percentage variation explained by leading eigenvectors, visual inspection of artificial phase space portraits, cross- and forward validation model selection were considered in the section. The use of phase space portraits is an extremely valuable visual method for identifying patterns (periodicity and trend) that are present in a time series. Phase space portraits of paired eigenvectors can be used to identify periodic behaviour as well as the period length (integer valued). When individual eigenvectors are plotted, trend components in a series can be identified, by identifying slowly varying eigenvectors.

This thesis made contributions towards computer intensive approaches to model selection in SSA. Algorithm 4.1 performs cross validation model selection and is a suggestion due to Venter (1998). Venter's (1998) original suggestion was adapted in this thesis to include a bias/variance tradeoff. Cross validation, however, was found to be CPU expensive and not really stable enough as model selection procedure in SSA.

Two powerful forward validation (FV) model selection procedures were proposed in Algorithm 4.2 and Algorithm 4.3 of Section 4.4.5. Both these algorithms apply the joint-horizon k-period-ahead formulation, Cadzow-signal reconstruction and incorporate a bias/variance trade-off. This resulted in the stabilisation of model selection. Both algorithms made considerable savings in CPU time possible and produce very similar model selection results. Algorithm 4.2 is favoured above Algorithm 4.3 as it does not require pre-inspection of phase space portraits, and is therefore closer to a fully automated model selection procedure. Another contribution of this thesis in the form of FORTRAN source code (SHONGO.F), is included in the appendix. This program can be used to perform forecasting and model selection (CV or FV).

The algorithms, mentioned in the previous paragraph, use forward validation. This necessitates the splitting of a time series into three non-overlapping series. The lengths of the training-, validation- and test sets are chosen arbitrarily. In the case of monthly data, it is recommended that the training series length be set as long as possible and that the length of the validation series be set equal to a full year's observations. It is also recommended that at least $k > 6$ -period-ahead forecasting be used during the FV validation process. A time series with a quarterly periodic variation can be treated similarly. For time series that do not contain periodic variation, it is also recommended that the training set length be as long as possible and that at least $k > 6$ -period-ahead forecasting be employed. The test set can be used to test out-of-sample forecasting accuracy of the selected model and therefore serves as an extra measure of model adequacy.

This study also found that Algorithms 4.2 and 4.3 produce more stable model selection results for series of length 100 and longer. The use of these algorithms for series that contain less than 60 observations is not recommended. For short series, visual inspection of the artificial phase space portraits is recommended.

It is *prima facie* that a structural change in a time series can influence model selection results, irrespective of the technique used. Chapter 3 of Golyandina *et al.* (2001) can be consulted for procedures that identify structural changes.

Section 4.4.6 considered methods for robustifying the forward validation model selection process. That section also outlined the complex nature of SVD robustification in the presence of outliers in the training series. It was recommended that outliers be identified and "corrected" before model selection. Algorithm 4.5 in Section 4.5 proposed an algorithmic approach to the identification of a single outlier.

A small scale Monte Carlo (MC) simulation was performed in Section 4.6. It was evident that the vastness of the broad class of functions that SSA can handle, can force a very sparse factorial design. It was, nevertheless, evident from the MC results that Algorithm 4.2 produced stable model selection results.

Chapter 5

INFERENCE IN SSA FORECASTING

5.1 Introduction

This Chapter briefly outlines inferential procedures that mainly find application in the SSA forecasting paradigm. The Chapter does not attempt to present an exhaustive study of inferential procedures. Only inferential procedures that are useful within the central theme of this thesis will therefore be considered.

Thus far, very little mention has been made about inferential procedures that can be applied in the SSA context, and especially in the forecasting paradigm. Chapter 2 highlighted two different model assumptions, viz. white noise and serially correlated noise models. It was evident from these assumptions that the construction of a joint distribution for an SSA model is analytically intractable. This places severe restrictions on the type of inferential procedures that are available in SSA. Fortunately, methods such as the bootstrap exist, and can be used when such circumstances are encountered.

Section 5.2 is devoted to the use of bootstrap methods to produce confidence limits for signal series. The section also addresses the important theme of producing *prediction intervals* (PI's) for forecasts, under an additive model assumption. Golyandina *et al.* (2001) also provide brief coverage in their work on bootstrap methods for producing *confidence intervals*. Their approach, however, differs from the approach proposed in Section 5.2.2.

Conclusions are reached and recommendations are made in Section 5.3.

5.2 Confidence and Prediction Intervals

This section considers the use of bootstrap methods for the construction of *confidence intervals* for signal series and *prediction intervals* (PI's) for forecasts, under an additive model assumption. Section 2.4 in Golyandina *et al*. (2001) is also devoted to *confidence intervals*. A classical treatment of bootstrap methods can be found in Efron and Tibshirani (1998).

It is important to point out the difference between confidence- and prediction intervals, before we proceed. The difference is clearly outlined in the paper by Chatfield (1993, p. 121), viz.

" *..."confidence interval" is usually applied to estimates of (fixed but unknown) parameters. In contrast, a prediction interval (PI) is an estimate of an (unknown) future value that can be regarded as a random variable at the time the forecast is made.*"

Golyandina *et al*. (2001) considers bootstrap methods to produce confidence intervals for future values of reconstructed signal series. We will consider the construction of prediction intervals for future values that we are producing forecasts for, under an additive model assumption. This is done in Section 5.2.2. According to a literature search this topic has not been dealt with in SSA. First, we will consider the construction of confidence intervals for the signal series.

5.2.1 Confidence intervals for the signal series

The following section considers the construction of confidence intervals for a signal series. It is assumed that the following model structure is under consideration

$$x_t = f_t + \varepsilon_t \ , \ \ t = 1,\ldots,N, \tag{5.1}$$

where $\{f_t\}_{t=1}^{N}$ is a signal series from the broad class defined by Buchstaber (1994) and therefore governed by an LRF. Furthermore, the noise series is uncorrelated and conforms to the assumption

$$\varepsilon_t \sim .\left(0,\ \sigma^2\right). \tag{5.2}$$

The true underlying signal series and noise distribution are not known in practice. In such circumstances one can resort to the use of bootstrap methods to construct confidence intervals for the signal series. This is now given as Algorithm 5.1 below.

Algorithm 5.1 Bootstrap confidence interval for a signal series.

(a) Load and transform an observed time series (if necessary).

(b) Assuming that the window length (τ) and number of leading eigenvectors (r) to use are known, a reconstructed signal series $\left\{\hat{f}_t\right\}_{t=1}^{N}$ is formed. This can be done by either an approximate series of Golyandina *et al.* (2001) or by a Cadzow-signal series (cf. Section 2.8.3 in Chapter 2).

(c) A residual series is formed by subtracting $\hat{f}_t$ in step (b) from the observed time series in step (a), i.e. forming

$$e_t = x_t - \hat{f}_t\ ,\ t = 1,\ldots,N\,. \tag{5.3}$$

(d) The mean of the residual series in (5.3) is subtracted, resulting in

$$e_t = e_t - \bar{e}\ ,\ t = 1,\ldots,N. \tag{5.4}$$

(e) A bootstrap loop of B iterations is started.

(f) Bootstrap time series are formed by sampling with replacement from the centred residual series in (5.4), i.e. forming

$$x_t^{*(i)} = \hat{f}_t + e_t^{*(i)}\ ,\ t = 1,\ldots,N;\ i = 1,\ldots,B. \tag{5.5}$$

(g) The same type of reconstructed signal (approximate or Cadzow) as in step (b) is produced using the bootstrap time series in step (f). Suppose that these series are denoted by

$$\left\{\hat{f}_t^{*(i)}\right\}_{t=1}^{N}, \quad i = 1,\ldots,B. \tag{5.6}$$

The algorithm returns to step (e) until all the bootstrap series in (5.6) have been reconstructed. The bootstrapped series are back-transformed, if a transformation was applied in step (a).

(h) Suppose that $(1-\alpha)$ represents the required level of confidence. The individual upper confidence limits at each of the signal series' points, are then given by

$$\left\{\hat{f}_1^*\right\}_B^{(1-\alpha/2)},\ldots,\left\{\hat{f}_N^*\right\}_B^{(1-\alpha/2)} \tag{5.7}$$

where $\left\{\hat{f}_t^*\right\}_B^{(1-\alpha/2)}$ is the $100(1-\alpha/2)\%$ percentile of the bootstrap series $\left\{\hat{f}_t^*\right\},\ t \in \{1,\ldots,N\}$.

Similarly, the lower confidence limits are given by

$$\left\{\hat{f}_1^*\right\}_B^{(\alpha/2)},\ldots,\left\{\hat{f}_N^*\right\}_B^{(\alpha/2)} \tag{5.8}$$

where $\left\{\hat{f}_t^*\right\}_B^{(\alpha/2)}$ is the $100(\alpha/2)\%$ percentile of the bootstrap series $\left\{\hat{f}_t^*\right\},\ t \in \{1,\ldots,N\}$.

Remark 5.1

(a) More sophisticated methods for constructing bootstrap prediction intervals, than simply using quantiles, are available (see Efron and Tibshirani, 1998). The present method sufficed for our needs.

(b) Golyandina *et al.* (2001) proposes the use of the approximate series during the construction of bootstrap confidence intervals for the signal series. It is proposed that their method be compared, as part of an empirical investigation, with the use of the Cadzow-signal series.

5.2.2 Prediction intervals for forecasts

According to Peña *et al.* (2001, p. 111)

"*A usual way to express the uncertainty in our forecast is to compute a prediction confidence interval, that is, an interval that will contain the future value we are forecasting...*".

A general overview of prediction intervals (PI's) can be found in Chatfield (1993). The latter article deals with general procedures for constructing PI's, which also include bootstrap methods. A general form for an (approximate) $100(1-\alpha)\%$ prediction interval for a k-period-ahead forecast made at time n, is given by

$$\hat{x}_n(k) \pm z_{\alpha/2}\sqrt{\operatorname{var}\left[e_n(k)\right]} \tag{5.9}$$

where $e_n(k) = x_{n+k} - \hat{x}_n(k)$ is the error of the forecast and $z_{\alpha/2}$ is the usual $100(\alpha/2)\%$ point from a standard normal distribution. Here it is assumed that normality holds approximately.

The purpose of this section is to propose a method for constructing PI's for forecasts that are generated in SSA. The following additive model assumption is made

$$x_t = f_t + \varepsilon_t \ , \ t = 1, \ldots, N, \tag{5.10}$$

where

$$\varepsilon_t \overset{i.i.d.}{\sim} .\left(0,\ \sigma^2\right). \tag{5.11}$$

Stine (1985) proposed a bootstrap technique for constructing prediction intervals for regression predictions. The proposed technique features in texts on bootstrap methodology (Shao and Tu, 1995, p. 305; Davison and Hinkley, 1999, pp. 285-286). We will apply Stine's (1985) method to our situation in Algorithm 5.2 below. Before doing this we first describe Stine's method.

According to Stine (1985), the following methodology can be followed to produce bootstrap PI's for regression predictions. Suppose that the linear model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{5.12}$$

is under consideration. The least squares parameter estimates are given in vectorised form by

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}. \tag{5.13}$$

Using the parameter estimates in (5.13), the residuals for the regression can be calculated as follows

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}. \tag{5.14}$$

Let the theoretical regression forecast be given by

$$Y_f = \mathbf{x}_f'\boldsymbol{\beta} + \varepsilon_f \tag{5.15}$$

and the error therefore be given by

$$\varepsilon_f = Y_f - \mathbf{x}'_f \boldsymbol{\beta}. \tag{5.16}$$

The regression forecast, in the case of sampled data, is given by

$$\hat{y}_f = \mathbf{x}'_f \hat{\boldsymbol{\beta}}. \tag{5.17}$$

According to Stine (1985, p. 1027) the bootstrap prediction error in the latter case is given by

$$D^*_f = Y^*_f - \mathbf{x}'_f \hat{\boldsymbol{\beta}}^*, \tag{5.18}$$

where the asterisk is used to signify bootstrap re-sampling. The distribution of (5.18) can be denoted by $G^*(.;\ f)$. A $100(1-\alpha)\%$ bootstrap PI for the prediction in (5.17) is then given by

$$\left[\hat{Y}_f + G^{*-1}(\alpha/2;\ f);\ \hat{Y}_f + G^{*-1}(1-\alpha/2;\ f)\right], \tag{5.19}$$

where $G^{*-1}(\alpha/2;\ f)$ and $G^{*-1}(1-\alpha/2;\ f)$ respectively denote the $100(\alpha/2)\%$ and $100(1-\alpha/2)\%$ percentage points of the empirical distribution featuring in (5.18).

The bootstrap replicates, featuring in (5.18), are obtained using the following

$$Y^*_f = \mathbf{x}'_f \hat{\boldsymbol{\beta}} + \varepsilon^*_f, \tag{5.20}$$

where ε^*_f represents bootstraps from the residuals featuring in (5.14). Bootstraps for the least squares parameter estimates

$$\hat{\boldsymbol{\beta}}^* = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}^* \tag{5.21}$$

are obtained by re-estimating the least squares parameter estimates each time a bootstrap sample was created using

$$\mathbf{Y}^* = \mathbf{X}'\hat{\boldsymbol{\beta}} + \boldsymbol{\varepsilon}^*, \tag{5.22}$$

where $\boldsymbol{\varepsilon}^* = \left(\varepsilon_1^*, \ldots, \varepsilon_N^*\right)'$ represent bootstrap replications from the residuals featuring in (5.14) and $\hat{\boldsymbol{\beta}}$ was calculated in (5.13).

The above procedure actually uses a bootstrap within a bootstrap approach. This is not too clear from Stine's (1985) article. The bootstrap within bootstrap approach is clearer from the following algorithm in Davison and Hinkley (1999, pp. 285-286). Note that their notation was changed to match the notation used in this thesis.

"***Algorithm 6.4 (Prediction in linear regression)***

```
For $r = 1, \ldots, R$,
    1. simulate responses $\mathbf{y}_r^*$ according to $\mathbf{y}_r^* = \mathbf{X}\hat{\boldsymbol{\beta}} + \boldsymbol{\varepsilon}^*$;
    2. obtain least squares estimates $\hat{\boldsymbol{\beta}}_r^* = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}_r^*$; then
    3. for $m = 1, \ldots, M$
        (a) sample $\varepsilon_{f,m}^*$ from $e_1 - \overline{e}, \ldots, e_n - \overline{e}$, and
        (b) compute prediction error $\delta_{rm}^* = \mathbf{x}_f'\hat{\boldsymbol{\beta}}_r^* - \left(\mathbf{x}_f'\hat{\boldsymbol{\beta}} + \varepsilon_{f,m}^*\right)$.
```

... More useful would be a $(1-2\alpha)$ *prediction interval for* Y_f*, for which we need the* α *and* $(1-\alpha)$ *quantiles* a_α *and* $a_{1-\alpha}$*, say, of prediction error* δ*. Then the prediction interval would have limits*

$$\hat{y}_f - a_{1-\alpha},\ \hat{y}_f - a_\alpha.$$

The exact, but unknown, quantiles are estimated by empirical quantiles of the pooled $\delta^* s$, *whose ordered values we denote by* $\delta^*_{(1)} \leq \ldots \leq \delta^*_{(RM)}$. *The bootstrap prediction limits are*

$$\hat{y}_f - \delta^*_{((RM+1)(1-\alpha))},\ \hat{y}_f - \delta^*_{((RM+1)\alpha)},$$

where $\hat{y}_f = \mathbf{x}'_f \hat{\boldsymbol{\beta}}$."

PI's for the SSA forecasts are considered next and an analogy is established. Suppose that an observed time series $\{x_t\}_{t=1}^{N}$ from the model in (5.10) is available and let $\{\hat{f}_t\}_{t=1}^{N}$ denote a reconstructed signal series. The analogy between this and the regression model is that the latter is equivalent to calculating the least squares parameter estimates in the regression model. This represents "truth" and also constitutes the major assumption underlying the bootstrap methodology, i.e. that the model is the correct one.

The residual series is then formed as follows

$$e_t = x_t - \hat{f}_t\ ,\ t = 1,\ldots,N. \tag{5.23}$$

A k-period-ahead forecast is generated, using one of the forecasting techniques described in Chapter 3, and is denoted by $\hat{f}_{N+k}$. Re-sampling with replacement takes place from the centred residual series in (5.23) to form the bootstrap time series

$$x^*_t = \hat{f}_t + \varepsilon^*_t\ ,\ t = 1,\ldots,N. \tag{5.24}$$

The bootstrap time series in (5.24) are used to construct bootstrap k-period-ahead forecasts $\hat{f}^*_{N+k}$ (using the same forecasting method to construct $\hat{f}_{N+k}$). The bootstrap forecast error is denoted by

$$D^*_{N+k} = x^*_{N+k} - \hat{f}^*_{N+k}, \tag{5.25}$$

where

$$x^*_{N+k} = \hat{f}_{N+k} + \varepsilon^*_{N+k} \,. \tag{5.26}$$

The bootstraps, featuring in (5.26), are constructed by re-sampling from the centred residuals in (5.23). It is clear that the original k-period-ahead forecast $\hat{f}_{N+k}$ is added to each re-sampled residual and this step is actually a bootstrap within a bootstrap.

Let $G^*(.;\ N+k)$ denote the (bootstrap) distribution of the forecast error featuring in (5.25). A $100(1-\alpha)\%$ bootstrap PI for the "unknown" future observation x_{N+k} is then given by

$$\left[\hat{f}_{N+k} + G^{*^{-1}}(\alpha/2;\ N+k);\ \hat{f}_{N+k} + G^{*^{-1}}(1-\alpha/2;\ N+k)\right], \tag{5.27}$$

where $G^{*^{-1}}(\alpha/2;\ N+k)$ and $G^{*^{-1}}(1-\alpha/2;\ N+k)$ are quantiles of G^*.

This proposal is given in Algorithm 5.2, below.

Algorithm 5.2 Bootstrap prediction intervals (PI's) for forecasts.

(a) Load and transform an observed time series (if necessary).

(b) Assuming that the window length (τ) and number of leading eigenvectors (r) to use are known, a reconstructed signal series $\left\{\hat{f}_t\right\}_{t=1}^{N}$ is formed. This can be done by either reconstructing an approximate series of Golyandina *et al.* (2001) or by reconstructing a Cadzow-signal series (cf. Section 2.8.3 in Chapter 2).

(c) The k-period-ahead forecast $\hat{f}_{N+k}$ is constructed using one of the forecasting techniques described in Chapter 3.

(d) A residual series is formed by subtracting the signal series in step (b) from the observed time series in step (a), i.e. forming

$$e_t = x_t - \hat{f}_t \ ,\ t = 1,\ldots,N \,. \tag{5.28}$$

(e) The mean of the residual series in (5.3) is subtracted, resulting in

$$e_t = e_t - \bar{e} \;,\; t = 1,\ldots,N. \tag{5.29}$$

(f) A bootstrap loop of B_1 iterations is started.

(g) Bootstrap time series are formed by sampling with replacement from the centred residual series in (5.29), and forming

$$x_t^{*(i)} = \hat{f}_t + e_t^{*(i)} \;,\; t = 1,\ldots,N;\; i = 1,\ldots,B_1. \tag{5.30}$$

(h) The same type of signal, i.e. (i) approximate or (ii) Cadzow-signal, as in step (b) is reconstructed using the bootstrap time series in step (g). Suppose that these bootstrap signal series are denoted by

$$\left\{\hat{f}_t^{*(i)}\right\}_{t=1}^{N} \;,\; i = 1,\ldots,B_1. \tag{5.31}$$

The same forecasting algorithm, as was used in step (c), is used to produce the forecasts

$$\hat{f}_{N+k}^{*(i)} \;,\; i = 1,\ldots,B_1. \tag{5.32}$$

(i) Another bootstrap loop of B_2 is started. This loop is inside the previous bootstrap loop.

(j) The following bootstraps of the k-period-ahead forecasts are formed

$$x_{N+k}^{*(j)} = \hat{f}_{N+k} + e^{*(j)},\; j = 1,\ldots,B_2, \tag{5.33}$$

where $e^{*(j)}$ is re-sampled from (5.29).

(k) The bootstraps of the forecast error are then formed as follows

$$D_{N+k}^{*(ij)} = x_{N+k}^{*(j)} - \hat{f}_{N+k}^{*(i)}\ ,\ i = 1,\ldots,B_1;\ j = 1,\ldots,B_2. \tag{5.34}$$

(l) Suppose that $(1-\alpha)$ represents the required level of confidence. The $100(1-\alpha)\%$ prediction interval is then formed as follows

$$\left[\hat{f}_{N+k} + \left\{D_{N+k}^{*}\right\}_{B_1B_2}^{(\alpha/2)};\ \hat{f}_{N+k} + \left\{D_{N+k}^{*}\right\}_{B_1B_2}^{(1-\alpha/2)}\right], \tag{5.35}$$

where $\left\{D_{N+k}^{*}\right\}_{B_1B_2}^{(\alpha/2)}$ and $\left\{D_{N+k}^{*}\right\}_{B_1B_2}^{(1-\alpha/2)}$ are empirical percentiles of the ordered prediction errors $\left\{D_{N+k}^{*}\right\}_{(1)},\ldots,\left\{D_{N+k}^{*}\right\}_{(B_1B_2)}$ which were formed in (5.34).

Remark 5.2

(a) It is important to note that the above algorithm produces bootstrap prediction intervals under the assumption that thc underlying model is of the form as denoted in (5.10). This implies that the noise component is assumed to be independent and identically distributed. The bootstrap replicates make use of this attribute when sampling with replacement from the centred residual series. If the noise component has (eg.) an autoregressive structure, changes must be made to incorporate such a structural assumption.

(b) The bootstrap method also assumes that the correct model has been fitted to the observed series. No structure should be left in the resulting residual series.

Two examples, using Algorithm 5.2, are supplied below to illustrate the use of bootstrap PI's for forecasts.

Example 5.1 Bootstrap PI's for the airline series.

The airline series was used for the purposes of this example. Recall that the series consisted of 12 years of monthly data. The first 11 years (Jan-1949 to Dec-1959) were used to reconstruct a Cadzow-signal series. The time series was log-transformed, prior to the signal reconstruction. A window length of $\tau = 60$ and the leading $r = 13$ eigenvectors were used to form the Cadzow-signal.

Algorithm 5.2 was used to produce 95% and 99% PI's for the forecasts, over the last twelve months (Jan-1960 to Dec-1960). Note that recurrent forecasts were generated. The observed series, recurrent forecasts, lower- and upper prediction limits are summarised in Table 5.1 and Table 5.2, below. Note that $B_1 = B_2 = 200$ bootstrap repetitions were used.

Table 5.1 95% bootstrap PI's for the airline series forecasts.

Date	Series	Recurrent Forecasts	Lower 95% limit	Upper 95% limit
Jan-1960	417	420.852	376.532	435.297
Feb-1960	391	383.736	358.135	405.880
Mar-1960	419	450.084	426.486	478.658
Apr-1960	461	462.617	429.090	487.529
May-1960	472	483.882	462.357	524.309
Jun-1960	535	588.844	576.542	657.296
Jul-1960	622	672.477	650.675	728.947
Aug-1960	606	666.281	633.088	710.512
Sep-1960	508	561.081	516.470	586.724
Oct-1960	461	492.262	453.764	521.056
Nov-1960	390	431.263	417.274	476.387
Dec-1960	432	465.941	439.719	494.449

Table 5.2 99% bootstrap PI's for the airline series forecasts.

Date	Series	Recurrent Forecasts	Lower 99% limit	Upper 99% limit
Jan-1960	417	420.852	409.011	473.893
Feb-1960	391	383.736	354.453	424.294
Mar-1960	419	450.084	396.607	474.728
Apr-1960	461	462.617	417.223	499.486
May-1960	472	483.882	432.400	506.597
Jun-1960	535	588.844	557.244	638.637
Jul-1960	622	672.477	612.611	733.398
Aug-1960	606	666.281	595.574	712.886
Sep-1960	508	561.081	501.291	600.096
Oct-1960	461	492.262	432.341	517.555
Nov-1960	390	431.263	393.994	460.738
Dec-1960	432	465.941	425.624	498.654

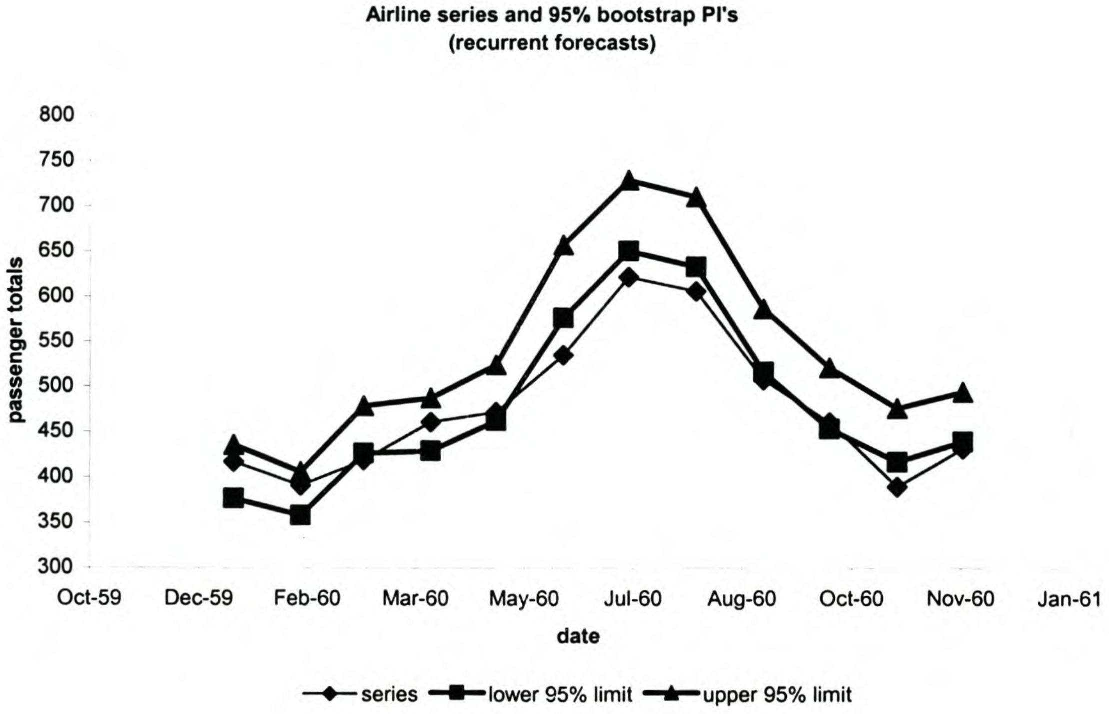


Figure 5.1 Airline series and 95% bootstrap PI's.

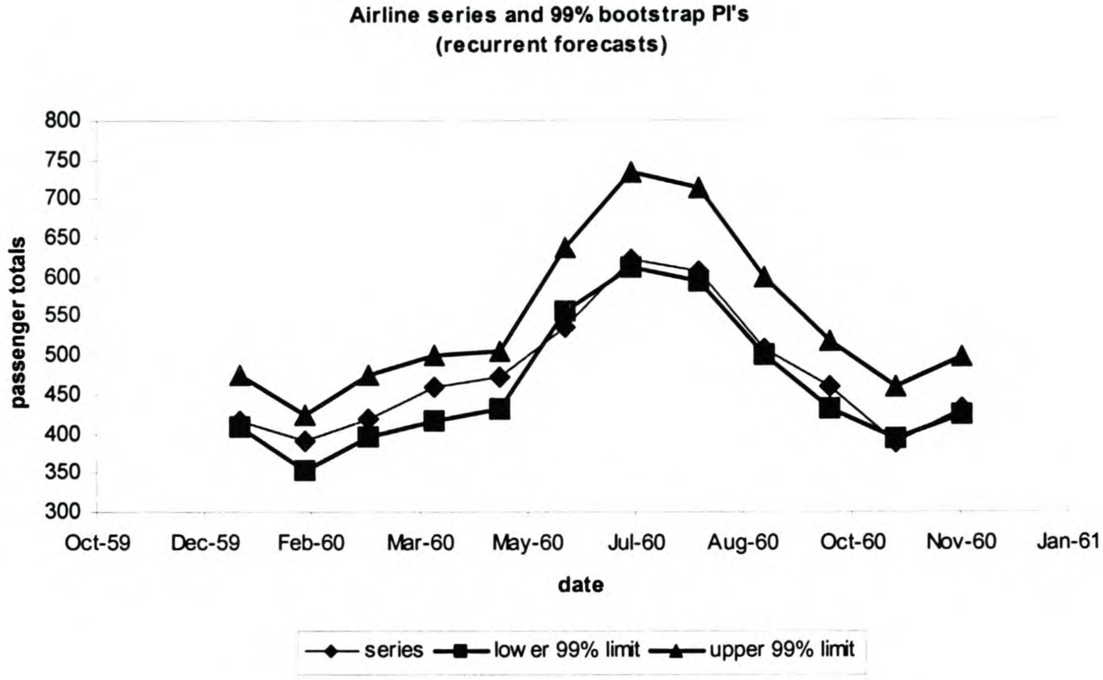


Figure 5.2 Airline series and 99% bootstrap PI's.

It is clear from perusal of Figure 5.1 and Figure 5.2 that the bootstrap PI's are relatively narrow. The 99% PI's seems to bound the observed series better than the 95% PI's. It is also clear that sometimes the prediction intervals do not include the observed values.

Example 5.2 Bootstrap PI's for the hotel series.
The hotel series was used for the purposes of this example. Recall that the series consisted of 14 years of monthly data. The first 13 years (Jan-1977 to Dec-1989) were used to reconstruct a Cadzow-signal series. The time series was not transformed, prior to the signal reconstruction. A window length of $\tau = 72$ and the leading $r = 12$ eigenvectors were used to form the Cadzow-signal.

Algorithm 5.2 was used to produce 95% and 99% bootstrap PI's for the recurrent forecasts, over the last twelve months (Jan-1990 to Dec-1990). The observed series, forecasts, lower- and upper prediction limits are summarised in Table 5.3 and Table 5.4, below.

The original series and 95% bootstrap PI's are graphically depicted in Figure 5.3, below. The PI's again seem to be on the narrow side and the coverage is not too good. Figure 5.4 depicts the 99% bootstrap PI's and the original series.

Table 5.3 95% bootstrap PI's for the hotel series forecasts.

Date	Series	Recurrent Forecasts	Lower 95% limit	Upper 95% limit
Jan-90	811	779.115	745.048	793.836
Feb-90	732	729.058	704.624	753.429
Mar-90	745	722.143	689.844	736.950
Apr-90	844	812.873	801.630	846.414
May-90	833	826.075	820.831	863.425
Jun-90	935	949.142	920.972	968.894
Jul-90	1110	1136.059	1108.699	1157.117
Aug-90	1124	1174.971	1156.379	1205.124
Sep-90	868	897.082	897.443	946.272
Oct-90	860	881.942	853.367	903.047
Nov-90	762	754.758	752.222	805.666
Dec-90	877	859.074	838.924	888.674

Table 5.4 99% bootstrap PI's for the hotel series forecasts.

Date	Series	Recurrent Forecasts	Lower 99% limit	Upper 99% limit
Jan-90	811	779.115	735.655	809.174
Feb-90	732	729.058	702.823	769.248
Mar-90	745	722.143	670.515	743.983
Apr-90	844	812.873	777.588	857.812
May-90	833	826.075	789.720	869.947
Jun-90	935	949.142	934.930	1001.674
Jul-90	1110	1136.059	1104.220	1184.409
Aug-90	1124	1174.971	1147.713	1228.004
Sep-90	868	897.082	871.000	944.485
Oct-90	860	881.942	832.002	912.160
Nov-90	762	754.758	731.850	805.368
Dec-90	877	859.074	816.862	890.330

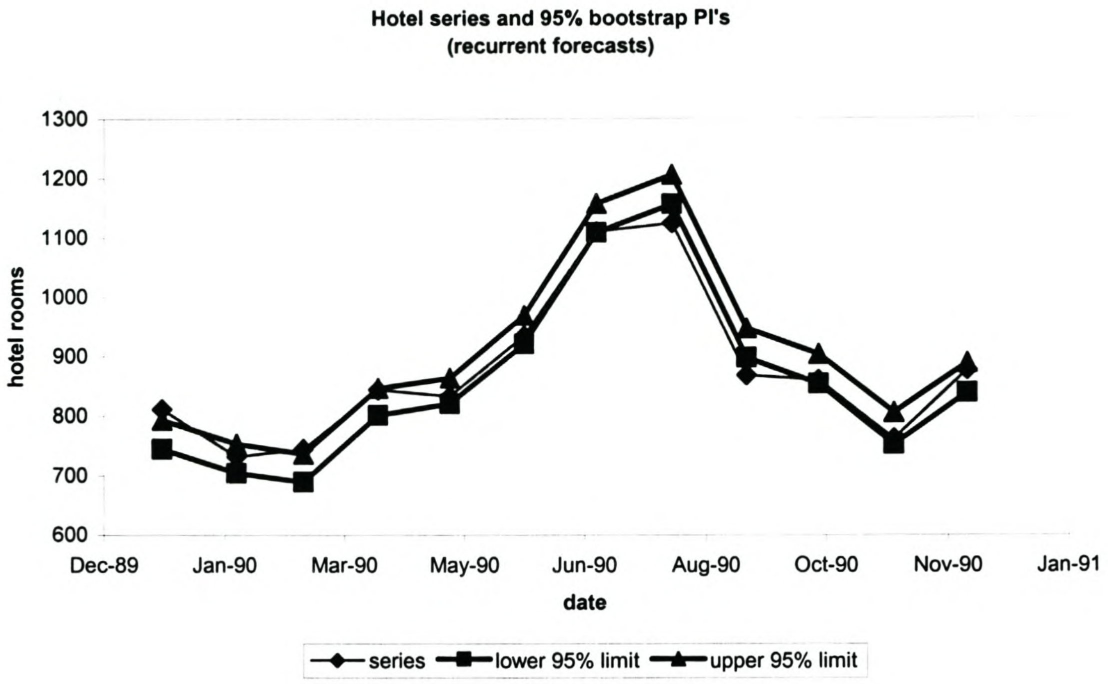


Figure 5.3 Hotel series and 95% bootstrap PI's.

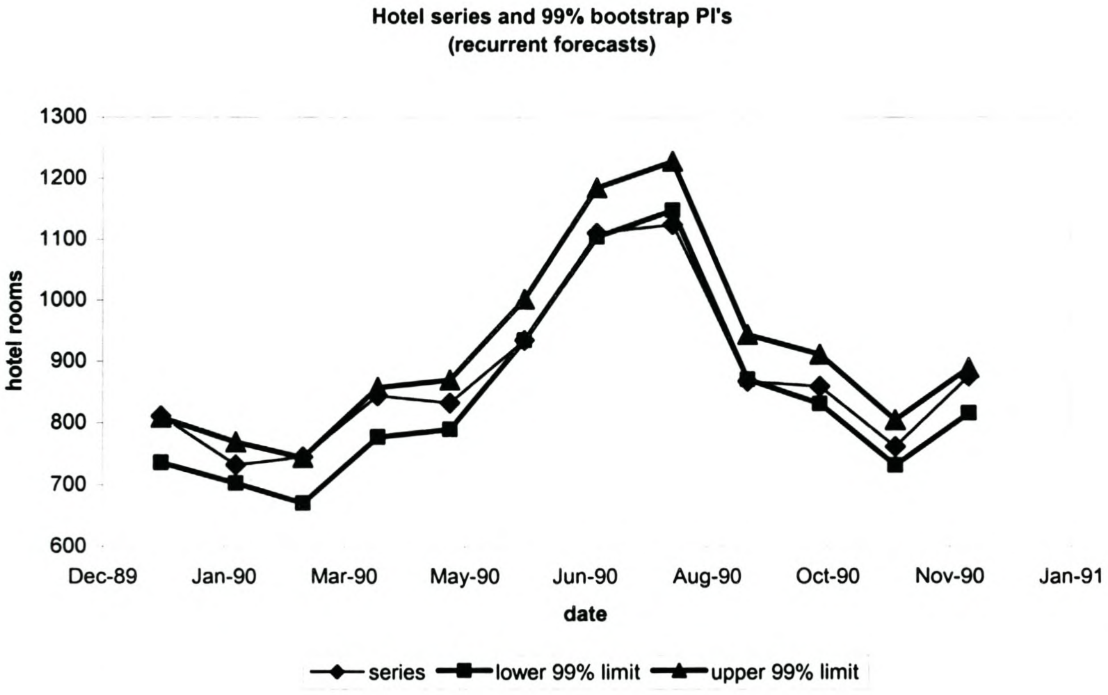


Figure 5.4 Hotel series and 99% bootstrap PI's.

5.3 Conclusions and Recommendations

This Chapter briefly considered inferential bootstrap procedures that find application in SSA forecasting.

Section 5.2 proposed an algorithm for the construction of confidence intervals for signal series and prediction intervals (PI's) for forecasts. The algorithm that was proposed for constructing PI's differs from the approach in Golyandina *et al.* (2001). Their work can be consulted in this regard. The use of bootstrap PI's make crucial assumptions, as is the case in Algorithm 5.2. It should be ensured that the assumptions of a model with additive independent and identically distributed noise structure hold, before employing algorithms proposed in this section. A correlogram, based on the residual series, can be inspected to ascertain that correlation or residual structure is not present in the residual series, which is bootstrapped from. If first-order serial correlation exists in the residuals, then the bootstrap must make use of this additional information and structural bootstrapping routines must be applied. The latter type of situation will be investigated in a future research project.

It was clear from the examples in this chapter that the proposed method for producing PI's did not seem to produce adequate intervals. The coverage performance of the intervals seemed to be inadequate. It is therefore recommended that a Monte Carlo simulation study be performed to investigate the coverage performance of PI's when using Algorithm 5.2, and that other possible methods also be investigated. In such a MC study, signal series from the broad class of functions proposed by Buchstaber (1994) would be considered. Different noise models and variances should be used. The effect on the coverage of PI's should be investigated when using different combinations of reconstructed signal series (approximate vs. Cadzow-signal) and forecasting methods (recurrent, vector and joint-horizon).

Chapter 6

SSA SOFTWARE

6.1 Introduction

The purpose of this Chapter is to report briefly on available software packages that perform singular spectrum analysis. The Chapter is not intended to produce an exhaustive elaboration on software packages, but rather to highlight features that are available.

Section 6.2 reports on the CaterpillarSSA version 3.00 software package. The software package was developed at St Petersburg University (Russia). An internet website address is provided where a trial version of the package can be downloaded. The main features of the package are also listed in the particular section.

Section 6.3 reports on a FORTRAN program, called SHONGO.F, which was developed as part of this study. Source code of this program is also included and can be found in the Appendix of this thesis. The main features of the program are listed in the section. Differences between SHONGO.F and the CaterpillarSSA software package are also listed in this section. SHONGO.F only uses algorithms that were developed as part of this research study. The reason for including SHONGO.F and this Chapter is to highlight features that this thesis has contributed to SSA. The new model selection features add value in the sense of automatic model selection that can possibly be incorporated in future commercial software packages.

6.2 CaterpillarSSA Version 3.00 Software Package

This software package was developed and is distributed by the GistaT Group (St Petersburg University, Russia). The group consists of Prof. Nina Golyandina, Prof. Vladimir Nekrutkin and Dr. Kirill Braulov. An internet website at which the software package and related literature can be found is available at address http://www.gistatgroup.com/cat/ .

The site allows downloading of the software for a 30-day trial period. The Caterpillar-SSA Version 3.00 has different editions as exhibited in Table 6.1 below.

Table 6.1 **Available editions of CaterpillarSSA Version 3.00 software.**

Edition	Module(s)
Standard Edition	1
Standard F Edition	1, 2
Standard M Edition	1, 4
Standard MF Edition	1, 2, 4
Professional Edition	1, 2, 3
Professional M Edition	1, 2, 3, 4

The functionality of each of the four modules in the CaterpillarSSA Version 3.00 software package is listed below, and is an excerpt of a help file. The current status, i.e. released version- or beta software, of the modules are reported in brackets.

Module 1. Analysis of one-dimensional time series (release)

- Decomposition of one-dimensional time series into eigentriples (eigenvalues, eigenvectors and principal components)
- Convenient graphical visualization of results for identification of the eigentriples corresponding to trend, periodicities, noise
- Grouping of eigentriples that leads to expansion of the time series into additive components
- Reconstruction of time series components (trend, oscillations, periodicities, noise) by choice of eigentriples
- Residual analysis

Module 2. Forecast of one-dimensional time series (release)

- Approximation (local) of time series by finite-rank series
- Forecast by vector and recurrent methods
- Analyzing the linear recurrent formula used for the recurrent forecast method
- Confidence intervals by empirical and bootstrap methods
- Testing the forecast results on validation period

Module 3. Change-point detection for one-dimensional time series (beta)

- Change-point detection by comparing the 'Caterpillar-SSA' structures of the base and test time series intervals
- Construction of heterogeneity matrix and detection functions
- Analyzing the found structural changes by moving root and modulus functions

Module 4. Processing multidimensional time series (beta)

- Simultaneous decomposition of several one-dimensional time series into common eigentriples (eigenvalues, eigenvectors and principal components)
- Convenient graphical visualization of results for identification of the eigentriples corresponding to common trend, periodicities and noise
- Grouping of eigentriples that leads to expansion of all the time series into additive components simultaneously
- Reconstruction of common time series components (trend, oscillations, periodicities, noise) by choice of eigentriples
- Approximation (local) of time series by finite-rank series
- Multivariate forecast by vector and recurrent methods
- Testing the forecast results on validation period

6.3 SHONGO.F FORTRAN Program

The FORTRAN source code of the program SHONGO.F is included in the Appendix of this thesis. This program uses only the algorithms derived as part of this study or formulations due to Venter (1998). A copy of the program can be obtained by contacting the author at e-mail address jdk2@akad.sun.ac.za.

The program does not have the same GUI (graphic user interface) as that which the CaterpillarSSA Version 3.00 software has. The program, however, has many additional features.

The SHONGO.F program can, *inter alia*, perform the following:

- Reconstruct an approximate series or a Cadzow-signal series
- Generate recurrent forecasts using the original series/ approximate series/ Cadzow-signal series
- Generate rolling recurrent one-period-ahead forecasts using the original series/ approximate series/ Cadzow-signal series
- Generate joint-horizon k-period-ahead forecasts using the original series/ approximate series/ Cadzow-signal series
- Perform cross validation (CV) model selection
- Perform forward validation (FV) model selection using the original series/ approximate series/ Cadzow-signal series
- Perform sequential grouping forward validation model selection using the original series/ approximate series/ Cadzow-signal series
- Perform batch mode forward validation model selection

Table 6.2, below, compares mathematical functionality of the programs. This study does not compare functionality such as database management, decomposition of time series and producing of graphical plots.

Table 6.2 **Comparison of mathematical functionality of programs.**

Functionality	**CaterpillarSSA Version 3.00 Software**	**SHONGO.F Program**
Allow choice between centring/ no centring of the trajectory matrix	Y	Y
Allow double centring of trajectory matrix	Y	N
Allow grouping of eigenvectors	Y	Y
Construct an approximate series	Y	Y
Construct Cadzow-signal series		Y
Produce in-sample recurrent forecasts	Y	Y
Produce out-of-sample recurrent forecasts	Y	Y
Produce in-sample k-period-ahead forecasts	Y (vector forecasting)	Y (joint-horizon)
Produce out-of-sample k-period-ahead forecasts	Y (vector forecasting)	Y (joint-horizon)
Produce out-of-sample measures of forecasting accuracy	N	MAD, MAPE, SSE, MSE
Perform CV model selection	N	Y
Perform FV model selection	N	Y
Perform sequential grouping FV model selection	N	Y
Perform batch mode FV model selection	N	Y
Produce confidence and prediction intervals	Y (bootstrap, empirical)	N

6.3.1 Sample output of forward validation model selection

This section illustrates the use of SHONGO.F. The section displays a screen dump of values that the user typically has to supply when performing sequential grouping forward validation model selection.

Option 3 was selected to perform sequential grouping forward validation model selection. The airline passenger series is used in this example. A natural logarithmic transformation was requested. The time series was split into three non-overlapping series, i.e. a training series $\{x_t\}_{t=1}^{120}$, validation series $\{x_t\}_{t=121}^{132}$ and test series $\{x_t\}_{t=133}^{144}$. The Cadzow-signal option was selected. This option reconstructs a Cadzow-signal during forward validation and forecasts are produced using this signal series. Note that 12-period-ahead forecasts were requested in this example, during FV. A maximum window length of $\tau_{\max} = 55$ was also requested. The following grouping matrix, used during the FV process, was supplied

$$\underset{8\times 3}{\mathbf{C}} = \left[\begin{array}{ccc} \#\ eigenvectors & Index\ 1 & Index\ 2 \\ \hat{1} & \hat{1} & \hat{} \\ 2 & 2 & 3 \\ 2 & 4 & 5 \\ 1 & 6 & \\ 1 & 7 & \\ 2 & 8 & 9 \\ 2 & 10 & 11 \\ 2 & 12 & 13 \end{array}\right].$$

The maximum number of eigenvectors, used during the FV process, was therefore 13 and the minimum window length to loop over was set at $\tau_{min} = k + r = 25$.

A screen dump of user supplied values and resulting output generated by SHONGO.F follows below.

Screen dump of supplied values

```
--------------------------------------------------
                Shongololo-SSA
                 developed by

                  J. de Klerk (2002)
  Department of Statistics and Actuarial Science
               Stellenbosch University
               Republic of South Africa
                email: jdk2@sun.ac.za
--------------------------------------------------
 The program uses forecasting algorithms that form
  part of the developer's PhD thesis, titled:-
    Time Series Forecasting and Model Selection
                      in
            Singular Spectrum Analysis
--------------------------------------------------
---------------- OPTIONS -------------------------
 Select an option from:-
  1. Reconstruct a Signal Series using SSA
  2. Forecast using SSA
  3. Forward Validation Model Selection
  4. Cross Validation Model Selection
3
------------- Forward Validation -----------------
 Type of FV
  1. Normal
  2. Sequential Grouping Forward Validation
  3. Batch Forward Validation
2
--------------------------------------------------
This program performs Forward Validation.
The program uses Sequential Grouping FV
Rolling k-period-ahead forecasts are used.
The technique uses centring of trajectory matrix
--------------------------------------------------
--- Transformation -------------------------------
Transformation to use 0=None,1=SQRT,2=LN
2
Time Series Filename       :
c:\phd\series\air.dat
Your TS is of length N=  144
--------------------------------------------------

---- Training Set --------------------------------
Train set length N1<=    144:
120
--------------------------------------------------
----- Validation Set -----------------------------
Validation set length N2<=      :  24
12
--------------------------------------------------
User series 0=Original, 1=Approx, 2=Cadzow:
2
k-period-ahead forecasts :
12
Window Length (maximum)  <=: 55
55
---- Sequential Grouping Information for FV -------
Number of groups to use
8
--------------------------------------------------
Number of Eigenvectors in group  1
1
Index value of eigenvector 1
1
Number of Eigenvectors in group  2
2
Index value of eigenvector 1
2
Index value of eigenvector 2
3
Number of Eigenvectors in group  3
2
Index value of eigenvector 1
4
Index value of eigenvector 2
5
Number of Eigenvectors in group  4
1
Index value of eigenvector 1
6
Number of Eigenvectors in group  5
```

```
1
Index value of eigenvector 1
7
Number of Eigenvectors in group  6
1
Index value of eigenvector 1
8
Number of Eigenvectors in group  7
2
Index value of eigenvector 1
10
Index value of eigenvector 2
11
Number of Eigenvectors in group  8
2
Index value of eigenvector 1
12
Index value of eigenvector 2
13
```

Screen dump of output in file

```
---------------------------------------------------
                 Shongololo-SSA
                  developed by

                   J. de Klerk (2002)
   Department of Statistics and Actuarial Science
                Stellenbosch University
                Republic of South Africa
                 email: jdk2@sun.ac.za
 ---------------------------------------------------
  The program uses forecasting algorithms that form
  part of the developer's PhD thesis, titled:-
     Time Series Forecasting and Model Selection
                       in
             Singular Spectrum Analysis
 ---------------------------------------------------
 ---------------------------------------------------------------------
  Sequential Grouping Forward Validation Results
 ---------------------------------------------------------------------
 ----------- Date of Analysis and Program Used -----------------------
 Date analysis started      : [ 16: 4:2002    ]
 Time analysis started      : [ 11:35:52      ]
 Running subroutine         : [      FVG2    ]
 ---------------------------------------------------------------------
 ----------- User supplied values ------------------------------------
 Transformation applied to series    : NLOG
 Series used                         : [ Cadzow-signal ]
 Window length (tau)       :[ 12<=tau<= 55]
 Time Series length (N)    :[144]
 Train series              :[   1... 120][ 120]
 Validation series         :[ 121... 132][  12]
 k-step-ahead forecasts (k):[ 12]
 ----------- Time Series Files ---------------------------------------
 File : c:\phd\series\air.dat
 ----------- Results -------------------------------------------------
 Forward validation results are in :
 C:\FV1.TXT
 ---------------------------------------------------------------------
 ----------- Grouping of Eigenvectors supplied -----------------------
       Group Eigenvectors      Index 1      Index 2
           1            1            1
           2            2            2            3
           3            2            4            5
           4            1            6
           5            1            7
           6            2            8            9
           7            2           10           11
           8            2           12           13
 ---------------------------------------------------------------------
 -------- Sequential Groupings used during FV ------------------------
 Model  Eigenvectors Used ("model")
     1 {[   1]}
     2 {[   1][  2,  3]}
     3 {[   1][  2,  3][  4,  5]}
     4 {[   1][  2,  3][  4,  5][  6]}
     5 {[   1][  2,  3][  4,  5][  6][  7]}
     6 {[   1][  2,  3][  4,  5][  6][  7][  8,  9]}
     7 {[   1][  2,  3][  4,  5][  6][  7][  8,  9][ 10, 11]}
     8 {[   1][  2,  3][  4,  5][  6][  7][  8,  9][ 10, 11][ 12, 13]}
 ---------------------------------------------------------------------
```

```
FV results sorted by mean(MSE)^2+var(MSE)
  Model   mean(MSE)              var(MSE)          mean(MSE)^2+var(MSE)
  8       408.847996767227       10547.2301743299  177703.914634905
  7       514.158006594273       10119.8151692982  274478.270914295
  6       634.141464665660       5757.21365856814  407892.610866877
  5       871.715805161024       85967.1083101164  845855.553277648
  3       985.444407878076       719.789477425133  971820.470495597
  2       1595.22674028130       1257.75010287102  2546006.10301136
  4       4170.09771215624       967924.899660944  18357639.8285917
  1       4685.71749379418       1476.10599728255  21957424.5376461

FV results sorted by var(MSE)
  Model   mean(MSE)              var(MSE)          mean(MSE)^2+var(MSE)
  3       985.444407878076       719.789477425133  971820.470495597
  2       1595.22674028130       1257.75010287102  2546006.10301136
  1       4685.71749379418       1476.10599728255  21957424.5376461
  6       634.141464665660       5757.21365856814  407892.610866877
  7       514.158006594273       10119.8151692982  274478.270914295
  8       408.847996767227       10547.2301743299  177703.914634905
  5       871.715805161024       85967.1083101164  845855.553277648
  4       4170.09771215624       967924.899660944  18357639.8285917
---------------------------------------------------------------------
Date analysis ended          : [ 16: 4:2002    ]
Time analysis ended          : [ 11:40:51      ]
---------------------------------------------------------------------
```

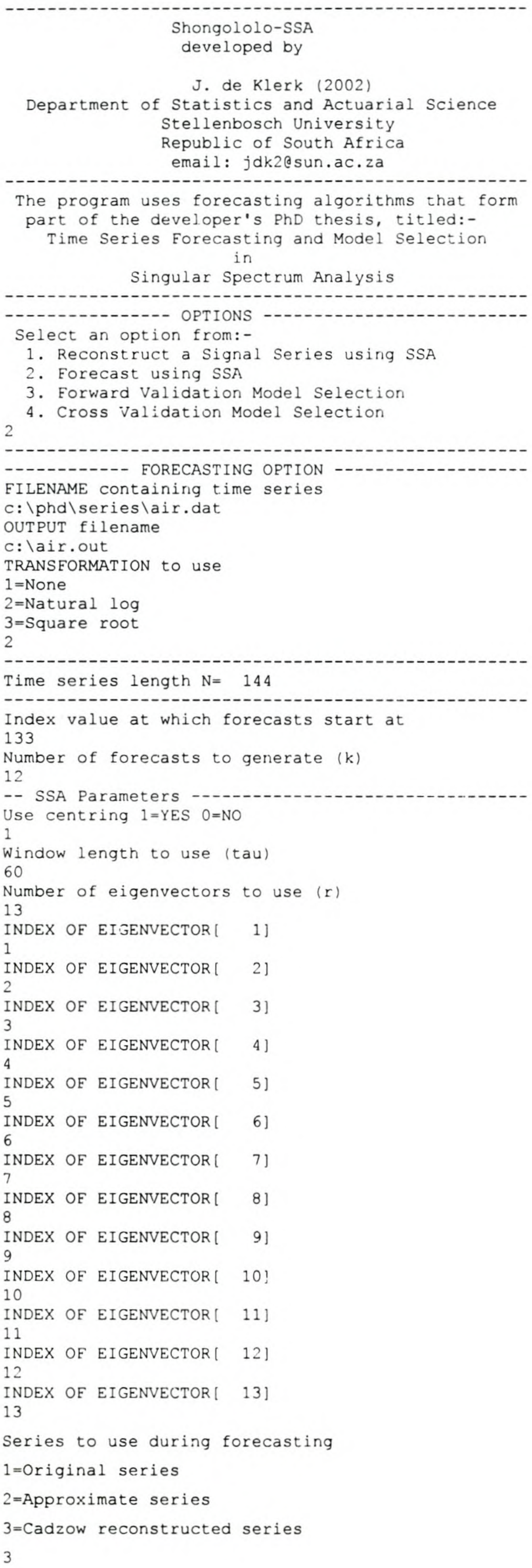
6.3.2 Sample output of forecasting

This section illustrates values that are usually required as input when producing forecasts with the SHONGO.F program.

The airline series was used in this example. A natural logarithmic transformation was applied to the time series. The program automatically back-transforms the series and forecasts, before calculating in-sample measures of forecasting accuracy. The length of the time series is 144 observations and the in-sample forecasts were set to start at time index $t = 133$. The number of forecasts to generate was set at 12. Centring of the trajectory matrix was requested. The window length was set at $\tau = 60$, which is an integer multiple of the monthly seasonal periodicity in the time series. The leading $r = 13$ eigenvectors were used to generate forecasts with. The Cadzow-signal series option was requested. This option resulted in the reconstruction of a Cadzow-signal series prior to the actual forecasting routine being called. The joint-horizon routine was then used to produce the in-sample forecasts with.

The screen dump of the information entered on-screen is supplied below. Below this screen dump, output generated by SHONGO.F can be found.

Screen dump



```
--------------------------------------------------
                Shongololo-SSA
                 developed by

                  J. de Klerk (2002)
  Department of Statistics and Actuarial Science
               Stellenbosch University
               Republic of South Africa
                email: jdk2@sun.ac.za
--------------------------------------------------
 The program uses forecasting algorithms that form
  part of the developer's PhD thesis, titled:-
    Time Series Forecasting and Model Selection
                      in
            Singular Spectrum Analysis
--------------------------------------------------
---------------- OPTIONS -------------------------
 Select an option from:-
  1. Reconstruct a Signal Series using SSA
  2. Forecast using SSA
  3. Forward Validation Model Selection
  4. Cross Validation Model Selection
2
--------------------------------------------------
------------ FORECASTING OPTION ------------------
FILENAME containing time series
c:\phd\series\air.dat
OUTPUT filename
c:\air.out
TRANSFORMATION to use
1=None
2=Natural log
3=Square root
2
--------------------------------------------------
Time series length N=  144
--------------------------------------------------
Index value at which forecasts start at
133
Number of forecasts to generate (k)
12
-- SSA Parameters --------------------------------
Use centring 1=YES 0=NO
1
Window length to use (tau)
60
Number of eigenvectors to use (r)
13
INDEX OF EIGENVECTOR[   1]
1
INDEX OF EIGENVECTOR[   2]
2
INDEX OF EIGENVECTOR[   3]
3
INDEX OF EIGENVECTOR[   4]
4
INDEX OF EIGENVECTOR[   5]
5
INDEX OF EIGENVECTOR[   6]
6
INDEX OF EIGENVECTOR[   7]
7
INDEX OF EIGENVECTOR[   8]
8
INDEX OF EIGENVECTOR[   9]
9
INDEX OF EIGENVECTOR[  10]
10
INDEX OF EIGENVECTOR[  11]
11
INDEX OF EIGENVECTOR[  12]
12
INDEX OF EIGENVECTOR[  13]
13
Series to use during forecasting
1=Original series
2=Approximate series
3=Cadzow reconstructed series
3
```

```
Forecasting technique to use
1=Recurrent one-period-ahead forecasts
2=Recurrent one-period-ahead forecasts re-estimated
3=Joint-horizon k-period-ahead forecasts
3
```

Output in File

```
---------------------------------------------------
                 Shongololo-SSA
                  developed by

                   J. de Klerk (2002)
   Department of Statistics and Actuarial Science
                Stellenbosch University
                Republic of South Africa
                 email: jdk2@sun.ac.za
 ---------------------------------------------------
  The program uses forecasting algorithms that form
  part of the developer's PhD thesis, titled:-
     Time Series Forecasting and Model Selection
                        in
             Singular Spectrum Analysis
 ---------------------------------------------------
 ----------- Time Series Information --------------------------------
 Date of analysis            : [ 24: 4:2002    ]
 Time of analysis            : [  9:56: 2      ]
 Running subroutine          : [ FORECASTING   ]
 Input Filename              : [  c:\phd\series\air.dat]
 Time Series length_N        : [           144 ]
 Window Length (tau)         : [            60 ]
 Transformation used         : [ Log  ]
 Centring used               : [           Yes ]
 Forecasted with series      : [Cadzow signal  ]
 Forecasting technique used  : [joint horizon  ]
 Forecasts starts at         : [           133 ]
 Number of forecasts         : [            12 ]
 Number of eigenvectors used: [            13 ]

 List of eigenvectors used   :
 Eigenvector[    1 ]         : [    1 ][  74.6530% ]
 Eigenvector[    2 ]         : [    2 ][   8.7087% ]
 Eigenvector[    3 ]         : [    3 ][   8.5837% ]
 Eigenvector[    4 ]         : [    4 ][   2.3686% ]
 Eigenvector[    5 ]         : [    5 ][   2.3175% ]
 Eigenvector[    6 ]         : [    6 ][   0.6647% ]
 Eigenvector[    7 ]         : [    7 ][   0.5427% ]
 Eigenvector[    8 ]         : [    8 ][   0.5300% ]
 Eigenvector[    9 ]         : [    9 ][   0.5126% ]
 Eigenvector[   10 ]         : [   10 ][   0.3413% ]
 Eigenvector[   11 ]         : [   11 ][   0.3315% ]
 Eigenvector[   12 ]         : [   12 ][   0.2250% ]
 Eigenvector[   13 ]         : [   13 ][   0.2207% ]
                                        [ 100.0000% ]
 FORECAST RESULTS
 ----------------------------------------------------------------
 Time                 Time Series                        Forecast
 ----------------------------------------------------------------
 133           417.000000000000               420.867219859946
 134           391.000000000000               383.749063848412
 135           419.000000000000               450.097134851178
 136           461.000000000000               462.630595095559
 137           472.000000000000               483.895642525574
 138           535.000000000000               588.859520735120
 139           622.000000000000               672.492208630595
 140           606.000000000000               666.293226882259
 141           508.000000000000               561.088233909474
 142           461.000000000000               492.266092108823
 143           390.000000000000               431.265238407176
 144           432.000000000000               465.942908013298
 ----------------------------------------------------------------
 MEASURES OF FORECASTING ACCURACY
 --------------------------------
 N    =                        12
 MAD =          31.6624130975492
 SSE =          16915.1827266064
 MSE =          1409.59856055054
 MAPE=                 6.4068771%
 --------------------------------
```

6.4 Conclusions and Recommendations

The Standard F edition of the CaterpillarSSA Version 3.00 software package was acquired and used during the course of this study. The software can install and run on any computer running Microsoft ® Windows (95, 98, Me, 2000 Professional, XP).

Electronic documentation accompanies the package, as is standard with modern software packages. The software is remarkably fast, when comparing it with the FORTRAN program developed as part of this study.

SHONGO.F can currently handle only univariate series of length 1000 and shorter. This is a shortcoming that can be overcome by using languages such as VB or VBA, which use dynamic array dimensioning. It must, however, be made clear, *ab initio*, that these languages are not nearly as fast as FORTRAN or C++. It is always possible to construct a GUI (graphics user interface) in one of the modern visual packages and use a high level language to handle mathematical operations behind the scenes. Information can then simply be supplied by the GUI package to DLL files, which were compiled by, for example, FORTRAN or C++.

It is envisaged that commercially available packages will in future have SSA as a standard option for time series analysis and forecasting. It is recommended that this process be completed as soon as possible, as SSA provides additional features to time series modelling.

Chapter 7

SUMMARY AND DIRECTIONS FOR FURTHER RESEARCH

This thesis contributed new techniques to the areas of forecasting and model selection in SSA. This Chapter summarizes the contributions made in the thesis, and also points out directions for further research.

Chapter 2 was devoted to the origins of the underlying elements of SSA. The main purpose of Chapter 2 was to introduce the groundbreaking advances that are due to Buchstaber (1994). The Chapter served as a foundation for the ensuing chapters. It is important to take note of the general class of functions, which can more generally be viewed as functions that are governed by linear recurrent formulae (LRF's), which SSA can handle.

Existing forecasting formulae were dealt with in Chapter 3. The Chapter considered formulae for producing recurrent-, joint-horizon k-period-ahead- and vector forecasts. The major contribution of Chapter 3, and this thesis, is a new general formulation for joint-horizon k-period-ahead forecasting. Recurrent one-period-ahead forecasting turned out to be a special case of the formula. The general joint-horizon k-period-ahead formula was also extended to the multi-channel SSA case. The Chapter also dealt with signal reconstruction in SSA. Two methods for signal reconstruction were considered, viz. using an approximate series (Golyandina *et al.*, 2001) or a reconstructed signal using the iterative mapping property of Cadzow (1988).

The Hankelization operation necessitated off-diagonal averaging of a projected matrix result. It became evident in Chapter 3 that signal reconstruction routines suffered from the same fate. The averaging was required to produce unique off-diagonal elements, which would in turn yield a reconstructed signal series. Signal reconstruction was performed to smooth out noise and was also necessary to yield a low-rank Hankel structured matrix, with column vectors on a single r-flat. It was mentioned that other techniques for signal reconstruction could possibly be researched. In particular, the structure total least norms (STLN) method was identified as a possible alternative (cf. Park *et al.*, 1999). The objective is to find a method that would produce a low-rank Hankel structured matrix with column vectors on a single r-flat of specified rank.

Model selection in SSA was considered in Chapter 4. The non-parametric nature of SSA pre-empted the use of non-parametric model selection techniques. Two computer intensive techniques, viz. cross- and forward validation, were considered. Recurrent one-period-ahead forecasting, during forward validation, was found not to result in stable model selection. It was shown that cross validation was more expensive on CPU time and did not take the dynamic nature of a time series into consideration, i.e. that more recent observations should carry more weight in the model selection process. The joint-horizon k-period-ahead forecasting formula, which was developed in Chapter 3, was successfully incorporated into a forward validation model selection technique. A major contribution of this thesis is a forward validation technique that uses signal reconstruction (Cadzow-signal), joint-horizon k-period-ahead forecasting combined with a bias-variance trade-off feature. Two forward validation model selection methods were developed, viz. one using leading eigenvectors and the other method using sequential grouping of eigenvectors. A small scale Monte Carlo simulation study was also conducted in Chapter 4, to illustrate the effectiveness of the developed forward validation model selection technique. Chapter 4 also considered robust procedures during forward validation and pointed out pitfalls. A method for the identification of outliers was also proposed.

Chapter 5, briefly, introduced two inferential procedures. Suggestions for confidence intervals for signal series and prediction intervals for forecasts, using bootstrap methodology, were made. Further research was identified in the case of prediction intervals. It was suggested that a Monte Carlo simulation study be used to compare the coverage probabilities of prediction intervals for forecasts that result from the use of different signal reconstruction methods.

FORTRAN source code, for the different procedures employed, was developed and is included in this thesis. The FORTRAN program SHONGO.F can perform recurrent one-period-ahead forecasting and joint-horizon k-period-ahead forecasting. Forecasts can be generated using the original time series, the approximate series or a Cadzow-signal series. The program can also perform cross validation model selection. Forward validation model selection is also possible. Two alternatives exist, viz. using leading eigenvectors or sequential grouping of eigenvectors. The forward validation model selection uses a bias-variance trade-off and a choice between using the approximate series or Cadzow-signal series, during forward validation, is also provided.

At the end of Chapter 3 it was also recommended that a simulation study be conducted to compare k-period-ahead forecasts produced by the recurrent one-period-ahead forecasting algorithms and that produced by the joint-horizon method. Such a study will confine itself to a few time series models and noise distributions.

It should be clear that this thesis covered a number of interesting topics in SSA, viz. signal reconstruction, model selection, outlier identification, forecasting- and inferential techniques. The literature study surely bares witness to the broad fields in which SSA has found application. It is expected that SSA will become a standard time series modelling and forecasting technique, somewhere in the not too distant future. This thesis also pointed out many pitfalls when applying SSA methodology. Many other research topics were identified during the course of this study, and offers scope for further timely research to be conducted in SSA…

APPENDIX

***Table A-1** International airline passengers: monthly totals (thousands of passengers) from January 1949 to December 1960*

Year / Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1949	112	118	132	129	121	135	148	148	136	119	104	118
1950	115	126	141	135	125	149	170	170	158	133	114	140
1951	145	150	178	163	172	178	199	199	184	162	146	166
1952	171	180	193	181	183	218	230	242	209	191	172	194
1953	196	196	236	235	229	243	264	272	237	211	180	201
1954	204	188	235	227	234	264	302	293	259	229	203	229
1955	242	233	267	269	270	315	364	347	312	274	237	278
1956	284	277	317	313	318	374	413	405	355	306	271	306
1957	315	301	356	348	355	422	465	467	404	347	305	336
1958	340	318	362	348	363	435	491	505	404	359	310	337
1959	360	342	406	396	420	472	548	559	463	407	362	405
1960	417	391	419	461	472	535	622	606	508	461	390	432

* Series obtained from Box, G.E.P. and Jenkins, G.M. (1970) Time Series Analysis: Forecasting and Control. San Francisco: Holden-day Publishing, p.531.

Table A-2 *Monthly hotel room averages for January 1977 to December 1990*

Year / Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1977	501	488	504	578	545	632	728	725	585	542	480	530
1978	518	489	528	599	572	659	739	758	602	587	497	558
1979	555	523	532	623	598	683	774	780	609	604	531	592
1980	578	543	565	648	615	697	785	830	645	643	551	606
1981	585	553	576	665	656	720	826	838	652	661	584	644
1982	623	553	599	657	680	759	878	881	705	684	577	656
1983	645	593	617	686	679	773	906	934	713	710	600	676
1984	645	602	601	709	706	817	930	983	745	735	620	698
1985	665	626	649	740	729	824	937	994	781	759	643	728
1986	691	649	656	735	748	837	995	1040	809	793	692	763
1987	723	655	658	761	768	885	1067	1038	812	790	692	782
1988	758	709	715	788	794	893	1046	1075	812	822	714	802
1989	748	731	748	827	788	937	1076	1125	840	864	717	813
1990	811	732	745	844	833	935	1110	1124	868	860	762	877

* Series obtained from Bowermann, B.L. and O'Connell, R.T. (1993) Forecasting and Time Series: An Applied Approach. California: Duxbury Press, p.312.

Table A-3 *Monthly liquor sales from January 1980 to December 1991*

Year / Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1980	480	467	514	505	534	546	539	541	551	537	584	854
1981	522	506	558	538	605	583	607	624	570	609	675	861
1982	605	537	575	588	656	623	661	668	603	639	669	915
1983	643	563	616	645	703	684	731	722	678	713	725	989
1984	687	629	687	706	754	774	825	755	751	783	804	1139
1985	711	693	790	754	799	824	854	810	798	807	832	1142
1986	740	713	791	768	846	884	886	878	813	840	884	1245
1987	796	750	834	838	902	895	962	990	882	936	997	1305
1988	866	805	905	873	1024	985	1049	1034	951	1010	1016	1378
1989	915	854	922	965	1014	1040	1137	1026	992	1052	1056	1469
1990	916	934	987	1018	1048	1086	1144	1077	1036	1076	1114	1595
1991	949	930	1045	1015	1091	1142	1182	1161	1145	1119	1189	1662
1992	1048	1019	1129	1092	1176	1297	1322	1330	1263	1250	1341	1927
1993	1271	1238	1283	1283	1413	1371	1425	1453	1311	1387	1454	1993

* Series obtained from Diebold, F.X. (1998) Elements of Forecasting

PROGRAM : SHONGO.F

EXPLANATION OF SUBROUTINES USED IN PROGRAM

SUBROUTINE	**TMAT(F, ITAU, ICOLS, X)**

Purpose

This subroutine creates the trajectory matrix (Hankel matrix structure)

$$\underset{(\tau \times n)}{\mathbf{X}} = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \\ f_2 & f_3 & \cdots & f_{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ f_\tau & f_{\tau+1} & \cdots & f_N \end{bmatrix}$$

by applying the rule

$$\begin{aligned} x_{i,j} &= f_{i+j-1} \\ i &= 1,\ldots,\tau\left(\leq \text{integer part of } \frac{\text{N+1}}{2}\right) \\ j &= 1,\ldots,n(= N - \tau + 1). \end{aligned}$$

In the above, τ is the window length and n the number of column vectors in trajectory matrix $\mathbf{X}$.

USER SUPPLIED VARIABLES

VARIABLE	USE
F	Vector containing the noise contaminated time series
ITAU	Window length (τ) to use (number of rows in matrix)
ICOLS	Number of columns in trajectory matrix $\mathbf{X}$

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
X	Trajectory matrix $\mathbf{X}$

SUBROUTINE	**TRAJECTORYMAT(F, ITAU, ICOLS, X)**

Purpose

This subroutine creates the trajectory matrix (Hankel matrix structure)

USER SUPPLIED VARIABLES

VARIABLE	USE
F	Vector containing the noise contaminated time series
ITAU	Window length (τ) to use (number of rows in matrix)
ICOLS	Number of columns in trajectory matrix $\mathbf{X}$

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
X	Trajectory matrix $\mathbf{X}$

SUBROUTINE	**CTMAT**(ITAU, ICOLS, X, XMEAN, ICENT, XTILDE)

Purpose

This subroutine creates centred trajectory matrix (Hankel matrix structure)

$$\begin{aligned}\underset{(T\times n)}{\tilde{\mathbf{X}}} &= \underset{(T\times n)}{\mathbf{X}} - \frac{1}{n}\mathbf{X}\mathbf{1}\mathbf{1}' \\ &= \underset{(T\times n)}{\mathbf{X}} - \bar{\mathbf{x}}\mathbf{1}'\end{aligned}$$

where

$$\begin{aligned}\underset{(n\times 1)}{\mathbf{1}} &= \begin{bmatrix}1 & \cdots & 1\end{bmatrix}' \\ \bar{\mathbf{x}} &= \text{shift vector of hyperplane } H.\end{aligned}$$

The subroutine makes provision for centring not to be applied when ICENT=0.

USER SUPPLIED VARIABLES

VARIABLE	USE
ITAU	Number of row vectors in trajectory matrix $\mathbf{X}$
ICOLS	Number of columns in trajectory matrix $\mathbf{X}$
X	Trajectory matrix $\mathbf{X}$
ICENT	Use centring if ICENT=1, else not

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
XMEAN	Shift vector ($\bar{\mathbf{x}}$) of r-flat H_r
XTILDE	Centred trajectory matrix $\tilde{\mathbf{X}}$

SUBROUTINE	**CTRAJECTORYMAT**(ITAU, ICOLS, X, XMEAN, XTILDE)

Purpose

This subroutine creates centred trajectory matrix (Hankel matrix structure). The subroutine uses centring by default.

USER SUPPLIED VARIABLES

VARIABLE	USE
ITAU	Number of row vectors in trajectory matrix $\mathbf{X}$
ICOLS	Number of columns in trajectory matrix $\mathbf{X}$
X	Trajectory matrix $\mathbf{X}$

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
XMEAN	Shift vector ($\bar{\mathbf{x}}$) of r-flat H_r
XTILDE	Centred trajectory matrix $\tilde{\mathbf{X}}$

SUBROUTINE	SCATMAT(XTILDE, ITAU, ICOLS, XXT)

Purpose

This subroutine creates the scatter matrix

$$\underset{(T\times T)}{\tilde{\mathbf{X}}\tilde{\mathbf{X}}'} = \left(\mathbf{X}-\bar{\mathbf{x}}\mathbf{1}'\right)\left(\mathbf{X}-\bar{\mathbf{x}}\mathbf{1}'\right)'$$

where

$$\underset{(n\times 1)}{\mathbf{1}} = \begin{bmatrix}1 & \cdots & 1\end{bmatrix}'$$
$$\bar{\mathbf{x}} = \text{shift vector of hyperplane } H.$$

The algorithm makes use of the fact that a scatter matrix is symmetrical. Only the diagonal and lower diagonal elements are constructed during the routine. The upper diagonal elements are set equal to the lower diagonal elements. This increases speed of programs that, during numeric intensive routines such as forward validation, uses this subroutine often.

USER SUPPLIED VARIABLES

VARIABLE	USE
XTILDE	Centred trajectory matrix $\tilde{\mathbf{X}}$
ITAU	Number of row vectors in trajectory matrix $\mathbf{X}$
ICOLS	Number of columns in trajectory matrix $\mathbf{X}$

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
XXT	Scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$

SUBROUTINE	PROJECTMAT(V, ITAU, IR, IEVEC, PV)

Purpose

This subroutine creates the projection matrix

$$\underset{(T\times T)}{\mathbf{P}_{\mathcal{L}_r}} = \underset{(T\times r)}{\mathbf{V}}\ \underset{(r\times T)}{\mathbf{V}'}$$

with matrix $\mathbf{V}$ the matrix with eigenvectors as column vectors, i.e.

$$\underset{(T\times r)}{\mathbf{V}} = \begin{bmatrix}\mathbf{v}_{c_1} & \mathbf{v}_{c_2} & \cdots & \mathbf{v}_{c_r}\end{bmatrix}$$

where $c_1, c_2, \ldots, c_r$ are the r index values of eigenvectors to use.

The eigenvectors are extracted from the scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$ using the double precision IMSL subroutine DEVCSF. The eigenvectors $\mathbf{v}_{c_1}, \mathbf{v}_{c_2}, \cdots, \mathbf{v}_{c_r}$ form an orthonormal base for parallel linear subspace $\mathfrak{L}_r$.

This subroutine also makes use of the fact that a projection matrix is symmetrical when constructing it. The diagonal and lower diagonal elements are constructed first. The upper diagonal elements are simply set equal to the lower diagonal elements. This aids in speeding up programs that are numerically intensive and uses this subroutine often.

USER SUPPLIED VARIABLES

VARIABLE	USE
V	Matrix with eigenvectors in columns
ITAU	Number of row vectors in trajectory matrix **V**
IR	Number of eigenvectors to use
IEVEC	Vector containing indices of eigenvectors to use

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
PV	Projection matrix $\mathbf{P}_{\mathfrak{L}_r}$ of parallel linear subspace $\mathfrak{L}_r$

SUBROUTINE	**PROJECTMAT2(**V, ITAU, IR, PV**)**

Purpose

This subroutine creates the projection matrix

$$\underset{(T\times T)}{\mathbf{P}_{\mathfrak{L}_r}} = \underset{(T\times r)}{\mathbf{V}}\ \underset{(r\times T)}{\mathbf{V}'}$$

with matrix **V** the matrix with eigenvectors as column vectors, i.e.

$$\underset{(T\times r)}{\mathbf{V}} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_r \end{bmatrix}.$$

The leading r eigenvectors are used to construct the projection matrix with.

USER SUPPLIED VARIABLES

VARIABLE	USE
V	Matrix with eigenvectors in columns
ITAU	Number of row vectors in trajectory matrix **V**
IR	Number of leading eigenvectors to use

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
PV	Projection matrix $\mathbf{P}_{\mathfrak{L}_r}$ of parallel linear subspace $\mathfrak{L}_r$

SUBROUTINE	**DEVCSF(**ITAU, XXT, IMAXTAU, EVAL, V, IMAXTAU**)**

Purpose

This is a double precision IMSL subroutine that is used to extract eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_\tau$ and eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_\tau$ from a covariance/scatter matrix.

USER SUPPLIED VARIABLES

VARIABLE	USE
ITAU	Dimension of scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$
XXT	Scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$
IMAXTAU	Maximum defined dimension of matrix XXT in program

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
EVAL	Vector of sorted eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_\tau$
V	Matrix with eigenvectors as column vectors

SUBROUTINE	**MEASURES(**INOBS, F, FHAT, MAD, MAPE, SSE, MSE**)**

Purpose

This subroutine calculates measure of forecasting accuracy. The measures of forecasting accuracy, calculated by this subroutine, are

$$MAD = \frac{1}{N}\sum_{t=1}^{N}\left|f_t - \hat{f}_t\right|$$

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\frac{\left|f_t - \hat{f}_t\right|}{f_t} \times 100\%$$

$$SSE = \sum_{t=1}^{N}\left(f_t - \hat{f}_t\right)^2$$

$$MSE = \frac{1}{N}SSE.$$

USER SUPPLIED VARIABLES

VARIABLE	USE
INOBS	Length of series to calculate measures for
F	Observed time series in a vector
FHAT	Forecasts in a vector

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
MAD	Mean Absolute Deviation
MAPE	Mean Absolute Percentage Error
SSE	Sum of Squared Errors
MSE	Mean Squared Error

SUBROUTINE	CADZOWSIGNAL(F, INOBS, ITAU, IR, IEVEC, ICENT, SIGNAL, XMEAN, PV, ITERATE)

Purpose

This subroutine reconstructs a signal series using the Cadzow (1988) iterative mapping algorithm. A univariate noise contaminated series is supplied to the subroutine. The user must also supply the window length (τ) and number of eigenvectors (r) that must be used. The actual indices of the eigenvectors to use are supplied in a vector.

LIMITATIONS OF SUBROUTINE

The program can reconstruct a single univariate time series' signal. The program can handle time series of length 1000 or shorter. Reconstructed signal series fall in the broad class of functions as describe in Chapter 2 of this thesis.

USER SUPPLIED VARIABLES

VARIABLE	USE
F	Vector containing noise contaminated time series
INOBS	Number of time series observations
ITAU	Window length (τ) to use
IR	Number of eigenvectors (r) to use
IEVEC	Vector containing indices of eigenvectors to use
ICENT	If ICENT=1 use centring, else not

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
SIGNAL	Vector containing Cadzow-signal series
XMEAN	Shift vector of Cadzow-signal series
PV	Projection matrix of Cadzow-signal series
ITERATE	Number of iterations to convergence of routine

SUBROUTINES CALLED BY THIS SUBROUTINE

SUBROUTINE
TMAT(F, ITAU, ICOLS, X)
CTMAT(ITAU, ICOLS, X, XMEAN ,ICENT, XTILDE)
SCATMAT(XTILDE, ITAU, ICOLS, XXT)
DEVCSF(ITAU, XXT, IMAXTAU, EVAL, V, IMAXTAU)
PROJECTMAT(V, ITAU, IR, IEVEC, PV)

EXPLANATION OF SUBROUTINE

The subroutine is iterative by nature. The observed time series is unfolded into a trajectory matrix. The centred trajectory matrix is formed (if ICENT=1). The scatter matrix is then formed. Eigenvectors are extracted from the scatter matrix. The projection matrix $\mathbf{P}_{\mathfrak{L}_r}$ is created using the eigenvectors with indices supplied in IEVEC. The following measure, which checks for convergence, is then calculated

$$\text{EPSILON} = \frac{\sum_{i=r+1}^{\tau} \lambda_i}{\sum_{i=1}^{r} \lambda_i}$$

with

$\lambda_i = ith$ eigenvalue of scatter matrix $\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$.

The condition for terminating the iterative loop is that $\text{EPSILON} < 10^{-9}$. If the condition is met, then the series at the current iteration is returned as the signal series, together with the shift vector and projection matrix.

If the above condition is not met then the following projection is performed,

$$\hat{\mathbf{X}} = \underbrace{\underline{\bar{\mathbf{x}}}\,\underline{\mathbf{1}}'}_{\text{shift back to hyperplane } H} + \underbrace{\mathbf{P}_{\mathfrak{L}_r}\tilde{\mathbf{X}}}_{\text{projection onto linear subspace } \mathfrak{L}_r}$$

where

$\hat{\mathbf{X}}$ = projected matrix $\mathbf{X}$ on current iteration's hyperplane.

A signal series, at the current iteration, is then reconstructed by averaging over the reverse diagonals of matrix $\hat{\mathbf{X}}$, using the Hankelization operation given by

$$\hat{f}_s = \begin{cases} \dfrac{1}{s}\sum_{i=1}^{s} \hat{x}_{i,s-i+1} & 1 \le s \le \tau \\ \dfrac{1}{\tau}\sum_{i=1}^{\tau} \hat{x}_{i,s-i+1} & \tau \le s \le n \\ \dfrac{1}{N-s+1}\sum_{i=1}^{N-s+1} \hat{x}_{i+s-n,n-i+1} & n \le s \le N \end{cases}$$

where

$\hat{f}_s$ = element s of reconstructed signal series (at current iteration)

$\hat{x}_{i,j}$ = element i, j of matrix $\hat{\mathbf{X}}$ (at current iteration).

The above procedure is repeated iteratively until the convergence criteria is met.

SUBROUTINE	**SIGNALSERIES(**F, INOBS, IN1, ITAU, ICOUNTR, SIGNAL, XMEAN, PV, ITERATE, ISERIES**)**

Purpose

This subroutine reconstructs a Cadzow-signal series if ISERIES=2. This subroutine differs from subroutine CADZOWSIGNAL in that the r leading eigenvectors are used and not specific ones.

If ISERIES=0 then the original series is returned with the shift vector and projection matrix. If ISERIES=1 then a single iteration is performed and the resulting signal series, shift vector and projection matrix is returned.

The subroutine also does not make provision for not using centring of the trajectory matrix. Centring of the trajectory matrix is used by default.

USER SUPPLIED VARIABLES

VARIABLE	USE
F	Vector containing noise contaminated time series
INOBS	Number of time series observations
IN1	Length of series to construct Cadzow-signal for
ITAU	Window length (τ) to use
ICOUNTR	Number of eigenvectors (r) to use
ISERIES	If ISERIES=0 no iteration used If ISERIES=1 a single iteration is used If ISERIES=2 the Cadzow-signal is returned

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
SIGNAL	Vector containing Cadzow-signal series
XMEAN	Shift vector of Cadzow-signal series
PV	Projection matrix of Cadzow-signal series
ITERATE	Number of iterations to convergence of routine

SUBROUTINES CALLED BY THIS SUBROUTINE

SUBROUTINE
TRAJECTORYMAT(F, ITAU, ICOLS, X**)**
CTMAT(ITAU, ICOLS, X, XMEAN, ICENT, XTILDE**)**
SCATMAT(XTILDE, ITAU, ICOLS, XXT**)**
DEVCSF(ITAU, XXT, IMAXTAU, EVAL, V, IMAXTAU**)**
PROJECTMAT(V, ITAU, IR, IEVEC, PV**)**

SUBROUTINE	**FORECAST(**F, IN1, IN2, ITAU, IR, IK, MAD, SSE, MSE, MAPE**)**

Purpose

This subroutine performs rolling 1-period-ahead forecasting over the validation series. The subroutine is used by the batch mode forward validation subroutine.

USER SUPPLIED VARIABLES

VARIABLE	USE
F	Vector containing noise contaminated time series
IN1	Length of training series
IN2	Length of validation series
ITAU	Window length (τ) to use
IR	Number of leading eigenvectors (r) to use
IK	Period-ahead forecasting to use

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
MAD	Mean Absolute Deviation over validation series
SSE	Sum of Squared Errors over validation series
MSE	Mean Square Error over validation series
MAPE	Mean Absolute Percentage Error over validation series

SUBROUTINES CALLED BY THIS SUBROUTINE

SUBROUTINE
TRAJECTORYMAT(F, ITAU, ICOLS, X**)**
CTRAJECTORYMAT(ITAU, ICOLS, X, XMEAN, XTILDE**)**
SCATMAT(XTILDE, ITAU, ICOLS, XXT**)**
DEVCSF(ITAU, XXT, IMAXTAU, EVAL, V, IMAXTAU**)**
PROJECTMAT2(V, ITAU, IR, PV**)**

SUBROUTINE	**APPROXSIGNAL(**F, INOBS, ITAU, IR, IEVEC, ICENT, SIGNAL, XMEAN, PV**)**

Purpose

This subroutine constructs an approximate signal series. It is equivalent to using a single iteration in the Cadzow subroutine. The subroutine allows the user to use only specific eigenvectors, by supplying the indices in vector IEVEC. Furthermore, centring of the trajectory matrix can be suppressed by setting ICENT=1.

Golyandina et al. (2001) uses the approximate signal in their work.

USER SUPPLIED VARIABLES

VARIABLE	USE
F	Vector containing noise contaminated time series
INOBS	Number of time series observations
ITAU	Window length (τ) to use
IR	Number of eigenvectors (r) to use
IEVEC	Vector containing indices of eigenvectors to use
ICENT	If ICENT=1 use centring, else not

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
SIGNAL	Vector containing approximate signal series
XMEAN	Shift vector of the approximate signal series
PV	Projection matrix of the approximate signal series

SUBROUTINE	**FORECAST3(**F, INOBS, ISTART, ITAU, IR, IEVEC, IK, ICENT, ICH1, FHAT, MAD, SSE, MSE, MAPE, ITRANS**)**

Purpose

This subroutine generates joint-horizon k-period-ahead forecasts. Forecasts can be generated using the original-, approximate signal or Cadzow-signal series. Centring of the trajectory matrix can be used or not. The subroutine also allows the user to specify the actual eigenvectors to use.

The subroutine checks the number of in-sample forecasts (if any) generated and returns the in-sample measures of forecasting accuracy MAD, SSE, MSE and MAPE.

The subroutine must be notified which type of transformation was applied to the time series. If ITRANS=1 then no transformation was used. If ITRANS=2 then a log transformation was used. If ITRANS=3 then a square root transformation was used. This information is required to back transform the results before calculating the measures of in-sample forecasting accuracy.

USER SUPPLIED VARIABLES

VARIABLE	USE
F	Vector containing noise contaminated time series
INOBS	Number of time series observations
ISTART	Index value at which forecasts start at
ITAU	Window length (τ) to use
IR	Number of eigenvectors (r) to use
IEVEC	Vector containing indices of eigenvectors to use
IK	Number of periods to forecast ahead
ICENT	If ICENT=1 use centring, else not
ICH1	If ICH1=1 use the original series to generate forecasts with If ICH1=2 use the approximate signal series to generate forecasts with If ICH1=3 use the Cadzow-signal series to generate forecasts with
ITRANS	If ITRANS=1 no transformation was applied to the series If ITRANS=2 then a log-transformation was applied to the series If ITRANS=3 then a square root transformation was applied to the series

SUBROUTINE	FORECAST2(F, INOBS, ISTART, ITAU, IR, IEVEC, IK, ICENT, ICH1, FHAT, MAD, SSE, MSE, MAPE, ITRANS)

Purpose

This subroutine produces rolling one-period-ahead forecasts. It has the same functionality as subroutine FORECAST3.

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
FHAT	Vector containing the forecasts
MAD	In-sample Mean Absolute Deviation
SSE	In-sample Sum of Squared Errors
MSE	In-sample Mean Squared Error
MAPE	In-sample Mean Absolute Percentage Error

SUBROUTINES CALLED BY THIS SUBROUTINE

SUBROUTINE
TMAT(F, ITAU, ICOLS, X**)**
CTMAT(ITAU, ICOLS, X, XMEAN, ICENT, XTILDE**)**
SCATMAT(XTILDE, ITAU, ICOLS, XXT**)**
DEVCSF(ITAU, XXT, IMAXTAU, EVAL, V, IMAXTAU**)**
PROJECTMAT(V, ITAU, IR, IEVEC, PV**)**

USER SUPPLIED VARIABLES

VARIABLE	USE
F	Vector containing noise contaminated time series
INOBS	Number of time series observations
ISTART	Index value at which forecasts start at
ITAU	Window length (τ) to use
IR	Number of eigenvectors (r) to use
IEVEC	Vector containing indices of eigenvectors to use
IK	Number of periods to forecast ahead
ICENT	If ICENT=1 use centring, else not
ICH1	If ICH1=1 use the original series to generate forecasts with If ICH1=2 use the approximate signal series to generate forecasts with If ICH1=3 use the Cadzow-signal series to generate forecasts with
ITRANS	If ITRANS=1 no transformation was applied to the series If ITRANS=2 then a log-transformation was applied to the series If ITRANS=3 then a square root transformation was applied to the series

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
FHAT	Vector containing the forecasts
MAD	In-sample Mean Absolute Deviation
SSE	In-sample Sum of Squared Errors
MSE	In-sample Mean Squared Error
MAPE	In-sample Mean Absolute Percentage Error

SUBROUTINES CALLED BY THIS SUBROUTINE

SUBROUTINE
TMAT(F, ITAU, ICOLS, X**)**
CTMAT(ITAU, ICOLS, X, XMEAN, ICENT, XTILDE**)**
SCATMAT(XTILDE, ITAU, ICOLS, XXT**)**
DEVCSF(ITAU, XXT, IMAXTAU, EVAL, V, IMAXTAU**)**
PROJECTMAT(V, ITAU, IR, IEVEC, PV**)**

SUBROUTINE	**FORECAST1(**F, INOBS, ISTART, ITAU, IR, IEVEC, IK, ICENT, ICH1, FHAT, MAD, SSE, MSE, MAPE, ITRANS**)**

Purpose

This subroutine produces one-period-ahead forecast without re-estimation. It also has the same functionality as subroutine FORECAST3.

USER SUPPLIED VARIABLES

VARIABLE	USE
F	Vector containing noise contaminated time series
INOBS	Number of time series observations
ISTART	Index value at which forecasts start at
ITAU	Window length (τ) to use
IR	Number of eigenvectors (r) to use
IEVEC	Vector containing indices of eigenvectors to use
IK	Number of periods to forecast ahead
ICENT	If ICENT=1 use centring, else not
ICH1	If ICH1=1 use the original series to generate forecasts with If ICH1=2 use the approximate signal series to generate forecasts with If ICH1=3 use the Cadzow-signal series to generate forecasts with
ITRANS	If ITRANS=1 no transformation was applied to the series If ITRANS=2 then a log-transformation was applied to the series If ITRANS=3 then a square root transformation was applied to the series

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
FHAT	Vector containing the forecasts
MAD	In-sample Mean Absolute Deviation
SSE	In-sample Sum of Squared Errors
MSE	In-sample Mean Squared Error
MAPE	In-sample Mean Absolute Percentage Error

SUBROUTINES CALLED BY THIS SUBROUTINE

SUBROUTINE
TMAT(F, ITAU, ICOLS, X)
CTMAT(ITAU, ICOLS, X, XMEAN, ICENT, XTILDE)
SCATMAT(XTILDE, ITAU, ICOLS, XXT)
DEVCSF(ITAU, XXT, IMAXTAU, EVAL, V, IMAXTAU)
PROJECTMAT(V, ITAU, IR, IEVEC, PV)

SUBROUTINE	**SIGNALSERIES2(**F, INOBS, IN1, ITAU, IR, SIGNAL, XMEAN, PV, ITERATE, ISERIES)

Purpose

This subroutine is used to return the original series with shift vector and projection matrix or the approximate signal with shift vector and projection matrix or Cadzow-signal series with shift vector and projection matrix. The subroutine uses the leading eigenvectors when constructing signals.

USER SUPPLIED VARIABLES

VARIABLE	USE
F	Vector containing noise contaminated time series
INOBS	Number of time series observations
IN1	Length of series to construct signal series for
ITAU	Window length (τ) to use
ICOUNTR	Number of eigenvectors (r) to use
ISERIES	If ISERIES=0 no iteration used If ISERIES=1 a single iteration is used If ISERIES=2 the Cadzow-signal is returned

SUBROUTINE RETURNED VARIABLES

VARIABLE	USE
SIGNAL	Vector containing the signal series
XMEAN	Shift vector of the signal series
PV	Projection matrix of the signal series
ITERATE	Number of iterations to convergence of routine

SUBROUTINE	**BATCHFVAL(**F, INOBS, IN1, IN2, IK, IR1, ITAU1, ISERIES, MSE1, MSE2, MSE3, ITAU2, IMODEL**)**

Purpose

This subroutine is used to perform batch mode forward validation. The subroutine is called by subroutine BATCH, which supplies all the required information.

The bias/variance trade-off measure, discussed in Chapter 4, is used during forward validation. Results are stored in file C:\OUT.TXT and C:\FV1.TXT.

SUBROUTINES CALLED BY THIS SUBROUTINE

SUBROUTINE
SIGNALSERIES2(F, INOBS, IN1, ITAU, IR, SIGNAL, XMEAN, PV, ITERATE, ISERIES**)**

SUBROUTINE	**BATCH**

Purpose

This subroutine obtains all the required information from the user to perform batch mode forward validation. Subroutine BATCHFVAL is then called.

SUBROUTINE	**FORECASTING**

Purpose

This subroutine obtains information from the user to perform forecasting. Depending on the user's choice one of three subroutines is then called.

A typical screen dump of user-supplied information will have the following appearance.

```
---------------------------------------------------
                Shongololo-SSA
                 developed by

                  J. de Klerk (2002)
  Department of Statistics and Actuarial Science
               Stellenbosch University
               Republic of South Africa
                email: jdk2@sun.ac.za
---------------------------------------------------
 The program uses forecasting algorithms that form
  part of the developer's PhD thesis, titled:-
    Time Series Forecasting and Model Selection
                      in
            Singular Spectrum Analysis
---------------------------------------------------
---------------- OPTIONS --------------------------
 Select an option from:-
  1. Reconstruct a Signal Series using SSA
  2. Forecast using SSA
  3. Forward Validation Model Selection
  4. Cross Validation Model Selection

2
---------------------------------------------------
------------ FORECASTING OPTION -------------------
FILENAME containing time series
c:\phd\series\air.dat
OUTPUT filename
c:\air.out
TRANSFORMATION to use
1=None
2=Natural log
```

```
3=Square root
2
---------------------------------------------------
Time series length N=  144
---------------------------------------------------
Index value at which forecasts start at
133
Number of forecasts to generate (k)
12
-- SSA Parameters ---------------------------------
Use centring 1=YES 0=NO
1
Window length to use (tau)
60
Number of eigenvectors to use (r)
13
INDEX OF EIGENVECTOR[    1]
1
INDEX OF EIGENVECTOR[    2]
2
INDEX OF EIGENVECTOR[    3]
3
INDEX OF EIGENVECTOR[    4]
4
INDEX OF EIGENVECTOR[    5]
5
INDEX OF EIGENVECTOR[    6]
6
INDEX OF EIGENVECTOR[    7]
7
INDEX OF EIGENVECTOR[    8]
8
INDEX OF EIGENVECTOR[    9]
9
INDEX OF EIGENVECTOR[   10]
10
INDEX OF EIGENVECTOR[   11]
11
INDEX OF EIGENVECTOR[   12]
12
INDEX OF EIGENVECTOR[   13]
13

Series to use during forecasting
1=Original series
2=Approximate series
3=Cadzow reconstructed series
3

Forecasting technique to use
1=Recurrent one-period-ahead forecasts
2=Recurrent one-period-ahead forecasts re-estimated
3=Joint-horizon k-period-ahead forecasts
3
```

The previous example used the airline passenger series. The series was log-transformed. The Cadzow-signal series was used during forecasting. Joint-horizon forecasting was requested. 12-period-ahead forecasts were requested starting at $t = 133$. Note that the 13 leading eigenvectors were supplied. The output file was named C:\AIR.OUT.

The output of file C:\AIR.OUT is exhibited below.

```
--------------------------------------------------
                 Shongololo-SSA
                  developed by

                 J. de Klerk (2002)
  Department of Statistics and Actuarial Science
               Stellenbosch University
               Republic of South Africa
                email: jdk2@sun.ac.za
 --------------------------------------------------
  The program uses forecasting algorithms that form
  part of the developer's PhD thesis, titled:-
     Time Series Forecasting and Model Selection
                        in
            Singular Spectrum Analysis
 --------------------------------------------------
 ----------- Time Series Information --------------------------------
 Date of analysis              : [ 24: 4:2002    ]
 Time of analysis              : [  9:56: 2      ]
 Running subroutine            : [ FORECASTING   ]
 Input Filename                : [  c:\phd\series\air.dat]
 Time Series length_N          : [            144 ]
 Window Length (tau)           : [             60 ]
 Transformation used           : [ Log   ]
 Centring used                 : [            Yes ]
 Forecasted with series        : [Cadzow signal ]
 Forecasting technique used : [joint horizon ]
 Forecasts starts at           : [            133 ]
 Number of forecasts           : [             12 ]
 Number of eigenvectors used: [             13 ]

 List of eigenvectors used   :
 Eigenvector[    1 ]          : [    1 ][  74.6530% ]
 Eigenvector[    2 ]          : [    2 ][   8.7087% ]
 Eigenvector[    3 ]          : [    3 ][   8.5837% ]
 Eigenvector[    4 ]          : [    4 ][   2.3686% ]
 Eigenvector[    5 ]          : [    5 ][   2.3175% ]
```

```
Eigenvector[    6 ]         : [   6 ][   0.6647% ]
Eigenvector[    7 ]         : [   7 ][   0.5427% ]
Eigenvector[    8 ]         : [   8 ][   0.5300% ]
Eigenvector[    9 ]         : [   9 ][   0.5126% ]
Eigenvector[   10 ]         : [  10 ][   0.3413% ]
Eigenvector[   11 ]         : [  11 ][   0.3315% ]
Eigenvector[   12 ]         : [  12 ][   0.2250% ]
Eigenvector[   13 ]         : [  13 ][   0.2207% ]
                                     [ 100.0000% ]

FORECAST RESULTS
------------------------------------------------------------------
Time                 Time Series                          Forecast
------------------------------------------------------------------
133             417.000000000000                  420.867219859946
134             391.000000000000                  383.749063848412
135             419.000000000000                  450.097134851178
136             461.000000000000                  462.630595095559
137             472.000000000000                  483.895642525574
138             535.000000000000                  588.859520735120
139             622.000000000000                  672.492208630595
140             606.000000000000                  666.293226882259
141             508.000000000000                  561.088233909474
142             461.000000000000                  492.266092108823
143             390.000000000000                  431.265238407176
144             432.000000000000                  465.942908013298
------------------------------------------------------------------
MEASURES OF FORECASTING ACCURACY
---------------------------------
N    =                         12
MAD  =           31.6624130975492
SSE  =           16915.1827266064
MSE  =           1409.59856055054
MAPE=                  6.4068771%
---------------------------------
```

SUBROUTINES CALLED BY THIS SUBROUTINE

SUBROUTINE
FORECAST1(F, INOBS, ISTART, ITAU, IR, IEVEC, IK, ICENT, ICH1, FHAT, MAD, SSE, MSE, MAPE, ITRANS**)**
FORECAST2(F, INOBS, ISTART, ITAU, IR, IEVEC, IK, ICENT, ICH1, FHAT, MAD, SSE, MSE, MAPE, ITRANS**)**
FORECAST3(F, INOBS, ISTART, ITAU, IR, IEVEC, IK, ICENT, ICH1, FHAT, MAD, SSE, MSE, MAPE, ITRANS**)**

SUBROUTINE | FVG2

Purpose

This subroutine obtains information from the user to perform sequential grouping forward validation, as outlined in Chapter 4. This forward validation subroutine uses the grouping idea of Golyandina et al. (2001). Groupings of eigenvectors are used rather than leading eigenvectors.

An example of a screen dump of user-supplied information is exhibited below.

```
---------------------------------------------------
                Shongololo-SSA
                 developed by

                  J. de Klerk (2002)
  Department of Statistics and Actuarial Science
               Stellenbosch University
               Republic of South Africa
                email: jdk2@sun.ac.za
---------------------------------------------------
 The program uses forecasting algorithms that form
  part of the developer's PhD thesis, titled:-
    Time Series Forecasting and Model Selection
                      in
            Singular Spectrum Analysis
---------------------------------------------------
---------------- OPTIONS --------------------------
 Select an option from:-
  1. Reconstruct a Signal Series using SSA
  2. Forecast using SSA
  3. Forward Validation Model Selection
  4. Cross Validation Model Selection
3
------------- Forward Validation ------------------
 Type of FV
  1. Normal
  2. Sequential Grouping Forward Validation
  3. Batch Forward Validation
2
---------------------------------------------------
This program performs Forward Validation.
The program uses Sequential Grouping FV
Rolling k-period-ahead forecasts are used.
The technique uses centring of trajectory matrix
---------------------------------------------------
--- Transformation --------------------------------
Transformation to use 0=None,1=SQRT,2=LN
2
Time Series Filename        :
c:\phd\series\air.dat
Your TS is of length N=  144
---------------------------------------------------

---- Training Set ---------------------------------
Train set length N1<=    144:
120
---------------------------------------------------
----- Validation Set ------------------------------
Validation set length N2<=      :  24
12
---------------------------------------------------
User series 0=Original, 1=Approx, 2=Cadzow:
2
k-period-ahead forecasts :
12
```

```
Window Length (maximum)  <=: 55
55
---- Sequential Grouping Information for FV -------
Number of groups to use
8
---------------------------------------------------
Number of Eigenvectors in group  1
1
Index value of eigenvector 1
1
Number of Eigenvectors in group  2
2
Index value of eigenvector 1
2
Index value of eigenvector 2
3
Number of Eigenvectors in group  3
2
Index value of eigenvector 1
4
Index value of eigenvector 2
5
Number of Eigenvectors in group  4
1
Index value of eigenvector 1
6
Number of Eigenvectors in group  5
1
Index value of eigenvector 1
7
Number of Eigenvectors in group  6
1
Index value of eigenvector 1
8
Number of Eigenvectors in group  7
2
Index value of eigenvector 1
10
Index value of eigenvector 2
11
Number of Eigenvectors in group  8
2
Index value of eigenvector 1
12
Index value of eigenvector 2
13
```

Output in the file C:\OUT.TXT is exhibited below.

```
----------------------------------------------------
                Shongololo-SSA
                 developed by

                J. de Klerk (2002)
  Department of Statistics and Actuarial Science
               Stellenbosch University
               Republic of South Africa
                email: jdk2@sun.ac.za
----------------------------------------------------
 The program uses forecasting algorithms that form
 part of the developer's PhD thesis, titled:-
    Time Series Forecasting and Model Selection
                        in
            Singular Spectrum Analysis
----------------------------------------------------
----------------------------------------------------------------------
 Sequential Grouping Forward Validation Results
----------------------------------------------------------------------
----------- Date of Analysis and Program Used -----------------------
Date analysis started       : [ 16: 4:2002    ]
Time analysis started       : [ 11:35:52      ]
Running subroutine          : [      FVG2    ]
----------------------------------------------------------------------
----------- User supplied values -------------------------------------
Transformation applied to series     : NLOG
Series used                          : [ Cadzow-signal ]
Window length (tau)        :[ 12<=tau<= 55]
Time Series length (N)     :[144]
Train series               :[    1... 120][ 120]
Validation series          :[ 121... 132][  12]
k-step-ahead forecasts (k):[ 12]
----------- Time Series Files ----------------------------------------
File : c:\phd\series\air.dat
----------- Results --------------------------------------------------
Forward validation results are in :
C:\FV1.TXT
----------------------------------------------------------------------
----------- Grouping of Eigenvectors supplied -----------------------
      Group Eigenvectors      Index 1      Index 2
          1            1            1
          2            2            2            3
          3            2            4            5
          4            1            6
          5            1            7
          6            2            8            9
          7            2           10           11
          8            2           12           13
----------------------------------------------------------------------
```

```
-------- Sequential Groupings used during FV -----------------------
Model  Eigenvectors Used ("model")
    1 {[  1]}
    2 {[  1][  2,  3]}
    3 {[  1][  2,  3][  4,  5]}
    4 {[  1][  2,  3][  4,  5][  6]}
    5 {[  1][  2,  3][  4,  5][  6][  7]}
    6 {[  1][  2,  3][  4,  5][  6][  7][  8,  9]}
    7 {[  1][  2,  3][  4,  5][  6][  7][  8,  9][ 10, 11]}
    8 {[  1][  2,  3][  4,  5][  6][  7][  8,  9][ 10, 11][ 12, 13]}
----------------------------------------------------------------------

FV results sorted by mean(MSE)^2+var(MSE)
  Model   mean(MSE)             var(MSE)            mean(MSE)^2+var(MSE)
  8       408.847996767227      10547.2301743299    177703.914634905
  7       514.158006594273      10119.8151692982    274478.270914295
  6       634.141464665660      5757.21365856814    407892.610866877
  5       871.715805161024      85967.1083101164    845855.553277648
  3       985.444407878076      719.789477425133    971820.470495597
  2       1595.22674028130      1257.75010287102    2546006.10301136
  4       4170.09771215624      967924.899660944    18357639.8285917
  1       4685.71749379418      1476.10599728255    21957424.5376461

FV results sorted by var(MSE)
  Model   mean(MSE)             var(MSE)            mean(MSE)^2+var(MSE)
  3       985.444407878076      719.789477425133    971820.470495597
  2       1595.22674028130      1257.75010287102    2546006.10301136
  1       4685.71749379418      1476.10599728255    21957424.5376461
  6       634.141464665660      5757.21365856814    407892.610866877
  7       514.158006594273      10119.8151692982    274478.270914295
  8       408.847996767227      10547.2301743299    177703.914634905
  5       871.715805161024      85967.1083101164    845855.553277648
  4       4170.09771215624      967924.899660944    18357639.8285917
----------------------------------------------------------------------
Date analysis ended          : [ 16: 4:2002    ]
Time analysis ended          : [ 11:40:51      ]
----------------------------------------------------------------------
```

SUBROUTINE FV

Purpose

This subroutine performs forward validation model selection. Chapter 4 can be consulted for the algorithm. The leading eigenvectors are used during forward validation.

SUBROUTINE	**CROSSV**

Purpose

This subroutine performs cross validation model selection. Consult Chapter 4 for the relevant algorithm.

FORTRAN SOURCE CODE

```
        PROGRAM SHONGOLOLOSSA
C ********************PROGRAM****************************************
C This program performs forecasting, forward- and cross validate
C model selection using SSA
C *********************PROGRAM BY***************************************
C                J. de Klerk [2002]
C **********************************************************************
      USE MSIMSL
C **********************************************************************
    1 FORMAT(A51)
      WRITE(*,1)'---------------------------------------------------'
      WRITE(*,1)'                Shongololo-SSA                     '
      WRITE(*,1)'                 developed by                      '
      WRITE(*,1)'                                                   '
      WRITE(*,1)'                  J. de Klerk (2002)               '
      WRITE(*,1)'  Department of Statistics and Actuarial Science   '
      WRITE(*,1)'               Stellenbosch University             '
      WRITE(*,1)'               Republic of South Africa            '
      WRITE(*,1)'                email: jdk2@sun.ac.za              '
      WRITE(*,1)'---------------------------------------------------'
      WRITE(*,1)' The program uses forecasting algorithms that form '
      WRITE(*,1)' part of the developer''s PhD thesis, titled:-    '
      WRITE(*,1)'    Time Series Forecasting and Model Selection    '
      WRITE(*,1)'                      in                           '
      WRITE(*,1)'            Singular Spectrum Analysis             '
      WRITE(*,1)'---------------------------------------------------'
      WRITE(*,1)'---------------- OPTIONS --------------------------'
      WRITE(*,1)' Select an option from:-                           '
      WRITE(*,1)'  1. Reconstruct a Signal Series using SSA         '
      WRITE(*,1)'  2. Forecast using SSA                            '
      WRITE(*,1)'  3. Forward Validation Model Selection            '
      WRITE(*,1)'  4. Cross Validation Model Selection              '
      WRITE(*,1)'                                                   '
      READ(*,*)ICHOICE1

C ** This section performs the SIGNAL RECONSTRUCTION **
      IF (ICHOICE1.EQ.1) THEN
       WRITE(*,1)'---------------------------------------------------'
       WRITE(*,1)'---------- SIGNAL RECONSTRUCTION ------------------'
       WRITE(*,1)' Type of signal series to reconstruct              '
       WRITE(*,1)'  1. Approximate series (single Cadzow-iteration)  '
       WRITE(*,1)'  2. Cadzow-signal series                          '
       READ(*,*)ICHOICE2
       CALL SIGRECONS(ICHOICE2)
      ENDIF

C ** This section performs the FORECASTING
      IF (ICHOICE1.EQ.2) THEN
       CALL FORECASTING
      ENDIF

C ** This section performs the FORWARD VALIDATION **
      IF (ICHOICE1.EQ.3) THEN
       WRITE(*,1)'------------- Forward Validation ------------------'
       WRITE(*,1)' Type of FV                                        '
       WRITE(*,1)'  1. Normal                                        '
       WRITE(*,1)'  2. Sequential Grouping Forward Validation        '
       WRITE(*,1)'  3. Batch Forward Validation                      '
       READ(*,*)ICHOICE2
       IF (ICHOICE2.EQ.1) THEN
        CALL FV
       ENDIF
       IF (ICHOICE2.EQ.2) THEN
        CALL FVG2
       ENDIF
       IF (ICHOICE2.EQ.3) THEN
        CALL BATCH
       ENDIF
      ENDIF

C ** This section performs the CROSS VALIDATION **
      IF (ICHOICE1.EQ.4) THEN
       CALL CROSSV
      ENDIF
      END

C *********************************************************************
C                 CROSSV [J. DE KLERK, 2002]
C *********************************************************************
      SUBROUTINE CROSSV
C *********************************************************************
C This subroutine performs Cross Validation Model Selection
C *********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      CHARACTER*60 FILENAME1,FILENAME2,FILENAME3,FILENAME4
      CHARACTER*4 TRANSFORM(3),MEASURE(3)
      CHARACTER*1 YESNO1,YESNO2
      DIMENSION F(IMAXOBS),SIGNAL(IMAXOBS),FHAT(IMAXOBS),FTEST(IMAXOBS)
      DIMENSION ORGF(IMAXOBS),X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION XL(IMAXTAU,IMAXCOLS),XMEANL(IMAXTAU)
      DIMENSION XXTL(IMAXTAU,IMAXTAU),VL(IMAXTAU,IMAXTAU)
      DIMENSION PVL(IMAXTAU,IMAXTAU),CV(IMAXTAU,IMAXTAU)
      DIMENSION XXT(IMAXTAU,IMAXTAU),V(IMAXTAU,IMAXTAU),EVAL(IMAXTAU)
      DIMENSION PV(IMAXTAU,IMAXTAU),XTILDE(IMAXTAU,IMAXCOLS)
      DIMENSION VEC1(IMAXTAU),VEC2(IMAXTAU),XHAT(IMAXOBS)
      DATA TRANSFORM/'NONE','SQRT','NLOG'/
      DATA MEASURE/'MAD','MAPE','MSE'/

      WRITE(*,*)'This program performs CROSS VALIDATION'
      WRITE(*,*)'The observations f(l) are left out one at a time.'
      WRITE(*,*)'The affected block of columns j1...j2 is not used'
      WRITE(*,*)'when constructing the mean vector and covariance'
      WRITE(*,*)'matrix. Columns j1 to j2 are reconstructed and'
      WRITE(*,*)'averaging over the reverse diagonals used to get'
      WRITE(*,*)'f(l) that was left out. The transformed series'
      WRITE(*,*)'with the f(l) in place is smoothed using CADZOW.'
      WRITE(*,*)'SIGNAL(l) is considered as reconstructed f(l).'

    1 FORMAT(A51)
C Get INPUT and OUTPUT filenames from user
      WRITE(*,1)'INPUT filename  :                                 '
      READ(*,*)FILENAME1
      WRITE(*,1)'OUTPUT filename :                                 '
      READ(*,*)FILENAME2
C Get transformation information from user
      WRITE(*,1)'Apply 1. NO 2. SQRT 3. LN transformation :        '
      READ(*,*)ITRANS
      OPEN(UNIT=1,FILE=FILENAME1,STATUS='OLD')
      OPEN(UNIT=2,FILE=FILENAME2)
```

```
C Read Time Series into vector F while counting the number of OBS
C Transform the original series if required by user
      INOBS=1
    2 READ(1,*,END=3)F(INOBS)
       ORGF(INOBS)=F(INOBS)
       IF (ITRANS.EQ.2) THEN
        F(INOBS)=DSQRT(F(INOBS))
       ENDIF
       IF (ITRANS.EQ.3) THEN
        F(INOBS)=DLOG(F(INOBS))
       ENDIF
       INOBS=INOBS+1
      GOTO 2
    3 INOBS=INOBS-1
      CLOSE(1)
      WRITE(*,*)'Loop tau from : '
      READ(*,*)ITAU1
      WRITE(*,*)'Loop tau to   : '
      READ(*,*)ITAU2
      WRITE(*,*)'Loop r from   : '
      READ(*,*)IR1
      WRITE(*,*)'Loop r to     : '
      READ(*,*)IR2
C Write information to OUTPUT FILE
      WRITE(2,*)'---------------------------------------------------'
      WRITE(2,*)'                  Shongololo-SSA                   '
      WRITE(2,*)'                   developed by                    '
      WRITE(2,*)'                                                   '
      WRITE(2,*)'                 J. de Klerk (2002)                '
      WRITE(2,*)'  Department of Statistics and Actuarial Science   '
      WRITE(2,*)'                Stellenbosch University            '
      WRITE(2,*)'                Republic of South Africa           '
      WRITE(2,*)'                 email: jdk2@sun.ac.za             '
      WRITE(2,*)'---------------------------------------------------'
      WRITE(2,*)' The program uses forecasting algorithms that form '
      WRITE(2,*)' part of the developer''s PhD thesis, titled:-    '
      WRITE(2,*)'    Time Series Forecasting and Model Selection    '
      WRITE(2,*)'                        in                         '
      WRITE(2,*)'              Singular Spectrum Analysis           '
      WRITE(2,*)'---------------------------------------------------'

      WRITE(2,*)'----------- Time Series Information -------------',
     +'---------------------'
      CALL TIMDY (IHOUR, IMIN, ISEC)
      CALL TDATE (IDAY, IMONTH, IYEAR)
      WRITE(2,4)'Date of analysis  Start     : [ ',IDAY,':',IMONTH,':',
     +IYEAR,'  ]'
      WRITE(2,5)'Time of analysis  Start     : [ ',IHOUR,':',IMIN,':',
     +ISEC,'    ]'
      WRITE(2,*)'Running Program             : [ CROSSVAL.F  ]'
    4 FORMAT (1X,A31,I2,A1,I2,A1,I4,A4)
    5 FORMAT (1X,A31,I2,A1,I2,A1,I2,A6)
      WRITE(2,6)'Input Filename              : [ ',FILENAME1,' ]'
    6 FORMAT(1X,A31,1X,A60,A2)
      WRITE(2,*)'Time Series length_N        : [  ',INOBS,']'
    7 FORMAT(A31,I2,A1,I2,A1)
      WRITE(2,7)'Loop structure for tau      : [',ITAU1,';',ITAU2,']'
      WRITE(2,7)'Loop structure for r        : [',IR1,';',IR2,']'
      WRITE(2,*)'----------- Notes -------------------------------',
     +'---------------------'
      WRITE(2,*)'Transformation applied to series    : ',
     +TRANSFORM(ITRANS)
      WRITE(2,*)'-------------------------------------------------',
     +'---------------------'
```

```
      WRITE(2,*)'CROSS VALIDATION RESULTS'
      WRITE(2,*)'tau   r              CV'

C Loop over the window length
      DO ITAU=ITAU1,ITAU2
C Set number of columns [ICOLS] in trajectory matrix XL
       ICOLS=INOBS-ITAU+1
C Construct trajectory matrix XL
       CALL TRAJECTORYMAT(F,ITAU,ICOLS,XL)
C Loop over the number of principal components to use
       DO IR=IR1,MIN(ITAU-1,IR2)
        CV(ITAU,IR)=0.0D0
C Loop over the observations to leave out f(l)
        DO IL=1,INOBS
C Calculate column vectors which contain left out observation
C These columns are from J1 (included) to J2 (included)
         J1=MAX(1,IL-ITAU+1)
         J2=MIN(IL,ICOLS)
C Construct the mean vector [XMEANL]
         DO I=1,ITAU
          XMEANL(I)=0.0D0
          DO J=1,ICOLS
C Ignore the colum vectors in XL that contain the left out f(l)
           IF ((J.LT.J1).OR.(J.GT.J2)) THEN
            XMEANL(I)=XMEANL(I)+XL(I,J)
           ENDIF
          ENDDO
          XMEANL(I)=XMEANL(I)/DFLOAT(ICOLS-(J2-J1+1))
         ENDDO
C Construct the variance-covariance matrix [XXTL]
         DO I=1,ITAU
          DO IK=I,ITAU
           XXTL(I,IK)=0.0D0
           DO J=1,ICOLS
C Ignore the column vectors in XL that contain the left out f(l)
            IF ((J.LT.J1).OR.(J.GT.J2)) THEN
             XXTL(I,IK)=XXTL(I,IK)+(XL(I,J)-XMEANL(I))*
     +        (XL(IK,J)-XMEANL(IK))
            ENDIF
           ENDDO
           XXTL(I,IK)=XXTL(I,IK)/DFLOAT(ICOLS-(J2-J1+1)-1)
C Use symmetry property to construct the covariance matrix
           XXTL(IK,I)=XXTL(I,IK)
          ENDDO
         ENDDO

C Calculate Eigenvectors of Matrix XXTL that forms
C orthonormal base for subspace Vr (IMSL routine)
         CALL DEVCSF(ITAU,XXTL,IMAXTAU,EVAL,VL,IMAXTAU)
C Determine the rank of matrix XXTL, i.e. non-zero eigenvalues
C if rank is less than IR (eigenvectors requested) adjust IR
         IRANK=0
         DO IIR=1,IR
          IF (EVAL(IIR).GT.1E-9) THEN
           IRANK=IRANK+1
          ENDIF
         ENDDO
         IF (IRANK.LT.1) THEN
          IRANK=1
         ENDIF
C Create Projection Matrix PV(L) of linear subspace Vr(L)
         CALL PROJECTMAT2(VL,ITAU,IRANK,PVL)
C Reconstruct columns that contain left out observation
         IVAL=0
         DO J=J1,J2
          IVAL=IVAL+1
          DO IJ=1,ITAU
           VEC1(IJ)=XL(IJ,J)-XMEANL(IJ)
           VEC2(IJ)=PVL(IJ,IL-J+1)
          ENDDO
          VEC1(IL-J+1)=0.0D0
          SUM1=0.0D0
          SUM2=0.0D0
          DO IIJ=1,ITAU
           SUM1=SUM1+VEC1(IIJ)*VEC2(IIJ)
           SUM2=SUM2+VEC2(IIJ)*VEC2(IIJ)
          ENDDO
          IF ((SUM1/(1-SUM2)).GT.(2*XMEAN(ITAU))) THEN
           XHAT(IVAL)=XMEANL(IL-J+1)
          ELSE
           XHAT(IVAL)=XMEANL(IL-J+1)+SUM1/(1-SUM2)
          ENDIF
         ENDDO
C Average over the reverse diagonals to reconstruct f(l)
C The reverse diagonal elements have been stored in vector XHAT
C IVAL is the number of these reverse diagonal elements
C Insert reconstructed value in the working time series and then smooth
         F(IL)=0.0D0
         DO I=1,IVAL
          F(IL)=F(IL)+XHAT(I)/IVAL
         ENDDO
C Use CADZOW to smooth the series
         CALL SIGNALSERIES2(F,INOBS,INOBS,ITAU,IR,SIGNAL,XMEAN,PV,
     +   ITERATE,2)
C Use signal(l) as estimate for f(l) and calculate squared error
         IF (ITRANS.EQ.1) THEN
          CV(ITAU,IR)=CV(ITAU,IR)+(ORGF(IL)-SIGNAL(IL))**2
         ENDIF
         IF (ITRANS.EQ.2) THEN
          CV(ITAU,IR)=CV(ITAU,IR)+(ORGF(IL)-SIGNAL(IL)**2)**2
         ENDIF
         IF (ITRANS.EQ.3) THEN
          CV(ITAU,IR)=CV(ITAU,IR)+(ORGF(IL)-DEXP(SIGNAL(IL)))**2
         ENDIF
C Reset the observation f(l) to original value
         IF (ITRANS.EQ.1) THEN
          F(IL)=ORGF(IL)
         ENDIF
         IF (ITRANS.EQ.2) THEN
          F(IL)=DSQRT(ORGF(IL))
         ENDIF
         IF (ITRANS.EQ.3) THEN
          F(IL)=DLOG(ORGF(IL))
         ENDIF
        ENDDO
 199    FORMAT(I3,2X,I3,2X,G30.15)
        WRITE(2,199)ITAU,IR,CV(ITAU,IR)/DFLOAT(INOBS)
        WRITE(*,*)'ITAU=',ITAU,' IR=',IR
       ENDDO
      ENDDO
      CALL TIMDY (IHOUR, IMIN, ISEC)
      CALL TDATE (IDAY, IMONTH, IYEAR)
      WRITE(2,4)'Date analysis ended        : [ ',IDAY,':',IMONTH,':',
     +IYEAR,'  ]'
      WRITE(2,5)'Time analysis ended        : [ ',IHOUR,':',IMIN,':',
     +ISEC,'    ]'
      CLOSE(2)
      END
```

```
C **********************************************************************
C                    FV [J. DE KLERK, 2002]
C **********************************************************************
      SUBROUTINE FV
C **********************************************************************
C This subroutine performs Forward Validation Model Selection
C **********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      CHARACTER*60 FILENAME1,FILENAME2
      CHARACTER*1 CH1
      CHARACTER*2 CH2
      DIMENSION F(IMAXOBS),SIGNAL(IMAXOBS),XMEAN(IMAXTAU)
      DIMENSION X(IMAXTAU,IMAXCOLS),XTILDE(IMAXTAU,IMAXCOLS)
      DIMENSION V(IMAXTAU,IMAXTAU)
      DIMENSION XXT(IMAXTAU,IMAXTAU),EVAL(IMAXTAU)
      DIMENSION FHAT(IMAXOBS),PV(IMAXTAU,IMAXTAU)
      DIMENSION P1(IMAXTAU,IMAXTAU),P2(IMAXTAU,IMAXTAU)
      DIMENSION P(IMAXTAU,IMAXTAU),P2TP2(200,200)
      DIMENSION P2TP1(IMAXTAU,IMAXTAU),P2TP2INV(200,200)
      DIMENSION P2TP2IP2TP1(IMAXTAU,IMAXTAU)
      DIMENSION VEC1(IMAXTAU)
      DIMENSION MSEMAT(IMAXTAU,IMAXTAU-1),SORTMSE(100,3)
      CHARACTER*4 TRANSFORM(3)
      DATA TRANSFORM/'NONE','SQRT','NLOG'/

    1 FORMAT(A51)
      WRITE(*,1)'---------------------------------------------------'
      WRITE(*,1)'This program performs Forward Validation.          '
      WRITE(*,1)'Rolling k-period-ahead forecasts are used.         '
      WRITE(*,1)'The technique uses centring of trajectory matrix   '
      WRITE(*,1)'---------------------------------------------------'

C Get type of transformation to apply to time series from user
      WRITE(*,'(A40)')'Transformation to use 0=None,1=SQRT,2=LN:'
      READ(*,*)ITRANS

C Get time series filename from user
      WRITE(*,'(A29)')'TS Filename                 : '
      READ(*,*)FILENAME1

C Read time series into vector F, counting the number of observations
      OPEN(UNIT=1,FILE=FILENAME1)
      INOBS=1
    2 READ(1,*,END=3)F(INOBS)
       IF (ITRANS.EQ.1) THEN
        F(INOBS)=DSQRT(F(INOBS))
       ENDIF
       IF (ITRANS.EQ.2) THEN
        F(INOBS)=DLOG(F(INOBS))
       ENDIF
       INOBS=INOBS+1
      GOTO 2
    3 INOBS=INOBS-1
      CLOSE(1)

      WRITE(*,'(A24,I4,A1)')'Your TS is of length N= ',INOBS
      WRITE(*,*)

C Get length of training series from user
      WRITE(*,'(A24,I4,A1)')'Train series length N1<=',INOBS,':'
      READ(*,*)IN1

C Get length of validation series from user
      WRITE(*,'(A33,I4)')'Validation series length N2<=   :',INOBS-IN1
      READ(*,*)IN2

C Get series to use during FV from user
      WRITE(*,'(A43)')'User series 0=Original, 1=Approx, 2=Cadzow:'
      READ(*,*)ISERIES

C Get k-step-ahead forecasts from user
      WRITE(*,'(A27)')'k-period-ahead forecasts : '
      READ(*,*)IK

C Get window length (tau) from user
      WRITE(*,'(A29,I2)')'Window Length (maximum)  <=: ',
     +FLOOR(DFLOAT((IN1-IK+2)/2))
      READ(*,*)ITAU1

C Get maximum number of leading eigenvectors to use from user
      WRITE(*,'(A29,I2)')'Number of eigenvectors 1<=r<= ',
     +INT(ITAU1-1)
      READ(*,*)IR1

C Write output to file
      OPEN(UNIT=11,FILE='C:\OUT.TXT');
      WRITE(11,*)'---------------------------------------------------'
      WRITE(11,*)'                 Shongololo-SSA                    '
      WRITE(11,*)'                  developed by                     '
      WRITE(11,*)'                                                   '
      WRITE(11,*)'                   J. de Klerk (2002)              '
      WRITE(11,*)'  Department of Statistics and Actuarial Science   '
      WRITE(11,*)'                Stellenbosch University            '
      WRITE(11,*)'                Republic of South Africa           '
      WRITE(11,*)'                 email: jdk2@sun.ac.za             '
      WRITE(11,*)'---------------------------------------------------'
      WRITE(11,*)' The program uses forecasting algorithms that form '
      WRITE(11,*)' part of the developer''s PhD thesis, titled:-     '
      WRITE(11,*)'    Time Series Forecasting and Model Selection    '
      WRITE(11,*)'                        in                         '
      WRITE(11,*)'             Singular Spectrum Analysis            '
      WRITE(11,*)'---------------------------------------------------'
      WRITE(11,*)'-------------------------------------------------',
     +'---------------------'
      WRITE(11,*)' Normal Forward Valıdation Results               '
      WRITE(11,*)'-------------------------------------------------',
     +'---------------------'
      WRITE(11,*)'----------- Date of Analysis and Program Used ---',
     +'---------------------'
      CALL TIMDY (IHOUR, IMIN, ISEC)
      CALL TDATE (IDAY, IMONTH, IYEAR)
   10 FORMAT (1X,A31,I2,A1,I2,A1,I4,A4)
   20 FORMAT (1X,A31,I2,A1,I2,A1,I2,A6)
      WRITE(11,10)'Date of analysis started   : [ ',IDAY,':',IMONTH,':',
     +IYEAR,'  ]'
      WRITE(11,20)'Time of analysis started   : [ ',IHOUR,':',IMIN,':',
     +ISEC,'    ]'
      WRITE(11,*)'Running subroutine         : [     FV       ]'
      WRITE(11,*)'-------------------------------------------------',
     +'---------------------'
```

```
      WRITE(11,*)'----------- User supplied values ----------------',
     +'---------------------'

      WRITE(11,*)'Transformation applied to series    : ',
     +TRANSFORM(ITRANS+1)
      IF (ISERIES.EQ.0) THEN
       WRITE(11,*)'Series used : [ Original ]'
      ENDIF
      IF (ISERIES.EQ.1) THEN
       WRITE(11,*)'Series used : [ Approximte ]'
      ENDIF
      IF (ISERIES.EQ.2) THEN
       WRITE(11,*)'Series used : [ Cadzow-signal ]'
      ENDIF
      WRITE(11,'(1X,A28,I3,A7,I3,A1)')
     +'Window length (tau)       :[',IR1+IK,'<=tau<=',ITAU1,']'
      WRITE(11,'(1X,A38,I3,A1)')
     +'Leading eigenvectors used :[  1<= r <=',IR1,']'
      WRITE(11,'(1X,A28,I3,A1)')'Time Series length (N)    :[',INOBS,']'
      WRITE(11,'(1X,A35,I4,A2,I4,A1)')
     +'Train series              :[   1...',IN1,'][',IN1,']'
      WRITE(11,'(1X,A28,I4,A3,I4,A2,I4,A1)')
     +'Validation series         :[',IN1+1,'...',IN1+IN2,'][',IN2,']'
      WRITE(11,'(1X,A28,I3,A1)')'k-step-ahead forecasts (k):[',IK,']'
      WRITE(11,*)'----------- Time Series Files -------------------',
     +'---------------------'
      WRITE(11,*)'File : ',FILENAME1
      WRITE(11,*)'----------- Results -----------------------------',
     +'---------------------'
      WRITE(11,*)'Forward validation results are in :'
      WRITE(11,*)'C:\FV1.TXT'
      WRITE(11,*)'-------------------------------------------------',
     +'---------------------'

C Open OUTPUT file
      OPEN(UNIT=2,FILE='C:\FV1.TXT')

C FORWARD VALIDATION STARTS HERE
      DO 103 ITAU=IR1+IK,ITAU1
       DO 102 IR=1,MIN(ITAU-IK,ITAU-1,IR1)
C Set position in original series to start at
        IN3=IN1-IK+1
        IT2=IN3
        IN4=IN3
C Initialize vector for use
        MSE=0.0D0
        MSEMAT(ITAU,IR)=0.0D0

        DO 101 I1=1,IN2
         WRITE(*,*)'I1=',I1,' ITAU=',ITAU,' IR=',IR

C The routine that is called does NOT use reconstructed Cadzow-signal
C The convergence criteria has been set to perform no iterations
C The subroutine is used to return the shift vector and projection matrix
         CALL SIGNALSERIES2(F,INOBS,IN3,ITAU,IR,SIGNAL,XMEAN,PV,
     &   ITERATE,ISERIES)

C Create projection matrix of orthogonal parallel linear subspace Vr
         DO I=1,ITAU
          DO J=I,ITAU
           IF (I.EQ.J) THEN
            P(I,J)=1.0D0-PV(I,J)
           ELSE
            P(I,J)=0.0D0-PV(I,J)
           ENDIF
           P(J,I)=P(I,J)
          ENDDO
         ENDDO

C Partition projection matrix P into P1 and P2
         DO I=1,ITAU
          DO J=1,ITAU
           IF (J.LE.(ITAU-IK)) THEN
            P1(I,J)=P(I,J)
           ENDIF
           IF (J.GE.(ITAU-IK+1)) THEN
            P2(I,J-(ITAU-IK))=P(I,J)
           ENDIF
          ENDDO
         ENDDO

C Create P2'P2 and invert the result
         DO I=1,IK
          DO J=1,IK
           P2TP2(I,J)=0.0D0
           DO IJ=1,ITAU
            P2TP2(I,J)=P2TP2(I,J)+P2(IJ,I)*P2(IJ,J)
           ENDDO
           P2TP2(J,I)=P2TP2(I,J)
          ENDDO
         ENDDO

C Set up inverse of P2TP2
         CALL DLINDS(IK,P2TP2,200,P2TP2INV,200)

C Create P2'P1 and invert the result
         DO I=1,IK
          DO J=1,(ITAU-IK)
           P2TP1(I,J)=0.0D0
           DO IJ=1,ITAU
            P2TP1(I,J)=P2TP1(I,J)+P2(IJ,I)*P1(IJ,J)
           ENDDO
          ENDDO
         ENDDO

C Create inv(P2'P2)P2TP1
         DO I=1,IK
          DO J=1,(ITAU-IK)
           P2TP2IP2TP1(I,J)=0.0D0
           DO IJ=1,IK
            P2TP2IP2TP1(I,J)=P2TP2IP2TP1(I,J)+P2TP2INV(I,IJ)*P2TP1(IJ,J)
           ENDDO
          ENDDO
         ENDDO

C Create k-step-ahead forecasts for the multiple time series
         ICOUNT3=0
         DO II=1,ITAU-IK
          ICOUNT3=ICOUNT3+1
          VEC1(ICOUNT3)=SIGNAL(IN3-ITAU+IK+II)-
     +    XMEAN(ICOUNT3)
         ENDDO

C The k-period-ahead forecasts are generated here
C ALL the k-period-ahead forecasts are used to construct the MSE with
         DO I=1,IK
          FHAT(I)=0.0D0
          DO J=1,(ITAU-IK)
```

```
           FHAT(I)=FHAT(I)+P2TP2IP2TP1(I,J)*VEC1(J)
          ENDDO
          FHAT(I)=XMEAN(ITAU-IK+I)-FHAT(I)
C Calculate MSE over validation series over all the k-period-ahead forecasts
          IF (ITRANS.EQ.0) THEN
           MSEMAT(ITAU,IR)=MSEMAT(ITAU,IR)+((F(IN4+I)-FHAT(I))**2)/
     +     DFLOAT(IN2*IK)
          ENDIF
          IF (ITRANS.EQ.1) THEN
           MSEMAT(ITAU,IR)=MSEMAT(ITAU,IR)+((F(IN4+I)**2-
     +     FHAT(I)**2)**2)/DFLOAT(IN2*IK)
          ENDIF
          IF (ITRANS.EQ.2) THEN
           MSEMAT(ITAU,IR)=MSEMAT(ITAU,IR)+((DEXP(F(IN4+I))-
     +     DEXP(FHAT(I)))**2)/DFLOAT(IN2*IK)
          ENDIF
         ENDDO
C Increase training series length by 1
         IN3=IN3+1
         IN4=IN4+1
  101   CONTINUE
C Write MSE to OUTPUT files
        WRITE(2,*)ITAU,' ',IR,' ',MSEMAT(ITAU,IR)
  102  CONTINUE
  103 CONTINUE

C Calculate mean(MSE) and var(MSE)
      DO I=1,IR1
       DO J=1,4
        SORTMSE(I,J)=0.0D0
       ENDDO
      ENDDO

      DO I=1,IR1
C Store number of leading eigenvectors in column [1] of SORTMSE matrix
       SORTMSE(I,1)=I
C Calculate mean(MSE) and store in column [2] of SORTMSE matrix
       DO J=IR1+IK,ITAU1
        SORTMSE(I,2)=SORTMSE(I,2)+MSEMAT(J,I)/DFLOAT(ITAU1-IR1-IK+1)
       ENDDO
C Calculate var(MSE) and store in column [3] of SORTMSE matrix
       DO J=IR1+IK,ITAU1
        SORTMSE(I,3)=SORTMSE(I,3)+((MSEMAT(J,I)-SORTMSE(I,2))**2)/
     +  DFLOAT(ITAU1-IR1-IK)
       ENDDO
C Calculate mean(MSE)^2+var(MSE) and store in column [4] of SORTMSE matrix
       SORTMSE(I,4)=SORTMSE(I,2)**2+SORTMSE(I,3)
      ENDDO

C Sort the results by mean(MSE)^2+var(MSE), i.e. by column [4] of SORTMSE matrix
      DO I=1,IR1
       DO J=I,IR1
        IF (SORTMSE(J,4).LT.SORTMSE(I,4)) THEN
         TEMP1=SORTMSE(I,1)
         TEMP2=SORTMSE(I,2)
         TEMP3=SORTMSE(I,3)
         TEMP4=SORTMSE(I,4)
         SORTMSE(I,1)=SORTMSE(J,1)
         SORTMSE(I,2)=SORTMSE(J,2)
         SORTMSE(I,3)=SORTMSE(J,3)
         SORTMSE(I,4)=SORTMSE(J,4)
         SORTMSE(J,1)=TEMP1
         SORTMSE(J,2)=TEMP2
         SORTMSE(J,3)=TEMP3
         SORTMSE(J,4)=TEMP4
        ENDIF
       ENDDO
      ENDDO

C Write sorted results to file
      WRITE(11,*)' '
      WRITE(11,*)'FV results sorted by mean(MSE)^2+var(MSE)'
      WRITE(11,'(3X,A1,11X,A9,22X,A8,23X,A20)')
     +'r','mean(MSE)','var(MSE)','mean(MSE)^2+var(MSE)'
      DO I=1,IR1
       WRITE(11,'(I4,3(A1,G30.15))')INT(SORTMSE(I,1)),' ',SORTMSE(I,2),
     + ' ',SORTMSE(I,3),' ',SORTMSE(I,4)
      ENDDO

C Sort the results by var(MSE), i.e. by column [3] of SORTMSE matrix
      DO I=1,IR1
       DO J=I,IR1
        IF (SORTMSE(J,3).LT.SORTMSE(I,3)) THEN
         TEMP1=SORTMSE(I,1)
         TEMP2=SORTMSE(I,2)
         TEMP3=SORTMSE(I,3)
         TEMP4=SORTMSE(I,4)
         SORTMSE(I,1)=SORTMSE(J,1)
         SORTMSE(I,2)=SORTMSE(J,2)
         SORTMSE(I,3)=SORTMSE(J,3)
         SORTMSE(I,4)=SORTMSE(J,4)
         SORTMSE(J,1)=TEMP1
         SORTMSE(J,2)=TEMP2
         SORTMSE(J,3)=TEMP3
         SORTMSE(J,4)=TEMP4
        ENDIF
       ENDDO
      ENDDO

C Write sorted results to file
      WRITE(11,*)' '
      WRITE(11,*)'FV results sorted by var(MSE)'
      WRITE(11,'(3X,A1,11X,A9,22X,A8,23X,A20)')
     +'r','mean(MSE)','var(MSE)','mean(MSE)^2+var(MSE)'
      DO I=1,IR1
       WRITE(11,'(I4,3(A1,G30.15))')INT(SORTMSE(I,1)),' ',SORTMSE(I,2),
     + ' ',SORTMSE(I,3),' ',SORTMSE(I,4)
      ENDDO

C Close the OUTPUT file
      CLOSE(2)

C Write the time and date that analysis ended to file
      CALL TIMDY (IHOUR, IMIN, ISEC)
      CALL TDATE (IDAY, IMONTH, IYEAR)
      WRITE(11,10)'Date analysis ended         : [ ',IDAY,':',IMONTH,':',
     +IYEAR,'  ]'
      WRITE(11,20)'Time analysis ended         : [ ',IHOUR,':',IMIN,':',
     +ISEC,'    ]'
      CLOSE(11)
      WRITE(*,*)'Comments and sorted results are in file C:\OUT.TXT'
      WRITE(*,*)'Pivot table data for EXCEL is in file C:\FV1.TXT'
      END
```

```
C *********************************************************************
C                   FVG2 [J. DE KLERK, 2002]
C *********************************************************************
c This subroutine gathers information to perform Sequential Grouping
C Forward Validation. The SGFV uses the Grouping idea in
C Golyandina et al. (2001)
C *********************************************************************
      SUBROUTINE FVG2
C *********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      CHARACTER*60 FILENAME1,FILENAME2
      CHARACTER*1 CH1
      CHARACTER*2 CH2
      CHARACTER*3 CH3
      CHARACTER*200 STRING1
      DIMENSION F(IMAXOBS),SIGNAL(IMAXOBS),XMEAN(IMAXTAU)
      DIMENSION X(IMAXTAU,IMAXCOLS),XTILDE(IMAXTAU,IMAXCOLS)
      DIMENSION XXT(IMAXTAU,IMAXTAU)
      DIMENSION FHAT(IMAXOBS),PV(IMAXTAU,IMAXTAU)
      DIMENSION P1(IMAXTAU,IMAXTAU),P2(IMAXTAU,IMAXTAU)
      DIMENSION P(IMAXTAU,IMAXTAU),P2TP2(200,200)
      DIMENSION P2TP1(IMAXTAU,IMAXTAU),P2TP2INV(200,200)
      DIMENSION P2TP2IP2TP1(IMAXTAU,IMAXTAU)
      DIMENSION VEC1(IMAXTAU),IC(30,3),IPC(40)
      DIMENSION MSEMAT(IMAXTAU,IMAXTAU-1),SORTMSE(100,3)
      CHARACTER*4 TRANSFORM(3)
      DATA TRANSFORM/'NONE','SQRT','NLOG'/

    1 FORMAT(A51)
      WRITE(*,1)'---------------------------------------------------'
      WRITE(*,1)'This program performs Forward Validation.          '
      WRITE(*,1)'The program uses Sequential Grouping FV            '
      WRITE(*,1)'Rolling k-period-ahead forecasts are used.         '
      WRITE(*,1)'The technique uses centring of trajectory matrix   '
      WRITE(*,1)'---------------------------------------------------'
      WRITE(*,1)'--- Transformation --------------------------------'
C Get type of transformation to apply to time series from user
      WRITE(*,'(A40)')'Transformation to use 0=None,1=SQRT,2=LN:'
      READ(*,*)ITRANS

C Get time series filename from user
      WRITE(*,'(A29)')'Time Series Filename       : '
      READ(*,*)FILENAME1

C Read time series into vector F, counting the number of observations
      OPEN(UNIT=1,FILE=FILENAME1)
      INOBS=1
    2 READ(1,*,END=3)F(INOBS)
       IF (ITRANS.EQ.1) THEN
        F(INOBS)=DSQRT(F(INOBS))
       ENDIF
       IF (ITRANS.EQ.2) THEN
        F(INOBS)=DLOG(F(INOBS))
       ENDIF
       INOBS=INOBS+1
      GOTO 2
    3 INOBS=INOBS-1
```

```
      CLOSE(1)

      WRITE(*,'(A24,I4,A1)')'Your TS is of length N= ',INOBS
      WRITE(*,1)'---------------------------------------------------'
      WRITE(*,*)

C Get length of training series from user
      WRITE(*,1)'---- Training Set ---------------------------------'
      WRITE(*,'(A24,I4,A1)')'Train set length N1<=   ',INOBS,':'
      READ(*,*)IN1
      WRITE(*,1)'---------------------------------------------------'

C Get length of validation series from user
      WRITE(*,1)'----- Validation Set ------------------------------'
      WRITE(*,'(A33,I4)')'Validation set length N2<=      :',INOBS-IN1
      READ(*,*)IN2
      WRITE(*,1)'---------------------------------------------------'

C Get series to use during FV from user
      WRITE(*,'(A43)')'User series 0=Original, 1=Approx, 2=Cadzow:'
      READ(*,*)ISERIES

C Get k-step-ahead forecasts from user
      WRITE(*,'(A27)')'k-period-ahead forecasts : '
      READ(*,*)IK

C Get window length (tau) from user
      WRITE(*,'(A29,I2)')'Window Length (maximum)  <=: ',
     +FLOOR(DFLOAT((IN1-IK+2)/2))
      READ(*,*)ITAU1

C Get number of groups to use
      WRITE(*,1)'---- Sequential Grouping Information for FV -------'
      WRITE(*,'(A29,I2)')'Number of groups to use      '
      READ(*,*)IGROUP
      WRITE(*,1)'---------------------------------------------------'

C Get the index values
      DO I=1,IGROUP
       WRITE(*,'(A32,I2)')'Number of Eigenvectors in group ',I
       READ(*,*)IC(I,1)
       DO J=1,IC(I,1)
        WRITE(*,'(A26,I2)')'Index value of eigenvector',J
        READ(*,*)IC(I,J+1)
       ENDDO
      ENDDO

C Write output to file
      OPEN(UNIT=11,FILE='C:\OUT.TXT');
      WRITE(11,*)'---------------------------------------------------'
      WRITE(11,*)'                  Shongololo-SSA                   '
      WRITE(11,*)'                   developed by                    '
      WRITE(11,*)'                                                   '
      WRITE(11,*)'                    J. de Klerk (2002)             '
      WRITE(11,*)'  Department of Statistics and Actuarial Science   '
      WRITE(11,*)'                 Stellenbosch University           '
      WRITE(11,*)'                 Republic of South Africa          '
      WRITE(11,*)'                  email: jdk2@sun.ac.za            '
      WRITE(11,*)'---------------------------------------------------'
      WRITE(11,*)' The program uses forecasting algorithms that form '
      WRITE(11,*)' part of the developer''s PhD thesis, titled:-     '
      WRITE(11,*)'    Time Series Forecasting and Model Selection    '
      WRITE(11,*)'                        in                         '
      WRITE(11,*)'              Singular Spectrum Analysis           '
      WRITE(11,*)'---------------------------------------------------'
      WRITE(11,*)'------------------------------------------------',
     +'---------------------'
      WRITE(11,*)' Sequential Grouping Forward Validation Results  '
      WRITE(11,*)'------------------------------------------------',
     +'---------------------'
      WRITE(11,*)'----------- Date of Analysis and Program Used ---',
     +'---------------------'
      CALL TIMDY (IHOUR, IMIN, ISEC)
      CALL TDATE (IDAY, IMONTH, IYEAR)
   10 FORMAT (1X,A31,I2,A1,I2,A1,I4,A4)
   20 FORMAT (1X,A31,I2,A1,I2,A1,I2,A6)
      WRITE(11,10)'Date analysis started      : [ ',IDAY,':',IMONTH,':',
     +IYEAR,'  ]'
      WRITE(11,20)'Time analysis started      : [ ',IHOUR,':',IMIN,':',
     +ISEC,'    ]'
      WRITE(11,*)'Running subroutine         : [     FVG2    ]'
      WRITE(11,*)'------------------------------------------------',
     +'---------------------'
      WRITE(11,*)'----------- User supplied values ----------------',
     +'---------------------'

      WRITE(11,*)'Transformation applied to series    : ',
     +TRANSFORM(ITRANS+1)
      IF (ISERIES.EQ.0) THEN
       WRITE(11,'(1X,A11,25X,A19)')'Series used',': [ Original ]     '
      ENDIF
      IF (ISERIES.EQ.1) THEN
       WRITE(11,'(1X,A11,25X,A19)')'Series used',': [ Approximte ]   '
      ENDIF
      IF (ISERIES.EQ.2) THEN
       WRITE(11,'(1X,A11,25X,A19)')'Series used',': [ Cadzow-signal ]'
      ENDIF
      WRITE(11,'(1X,A28,I3,A7,I3,A1)')
     +'Window length (tau)        :[',IR1+IK,'<=tau<=',ITAU1,']'
      WRITE(11,'(1X,A28,I3,A1)')'Time Series length (N)    :[',INOBS,']'
      WRITE(11,'(1X,A35,I4,A2,I4,A1)')
     +'Train series               :[   1...',IN1,'][',IN1,']'
      WRITE(11,'(1X,A28,I4,A3,I4,A2,I4,A1)')
     +'Validation series          :[',IN1+1,'...',IN1+IN2,'][',IN2,']'
      WRITE(11,'(1X,A28,I3,A1)')'k-step-ahead forecasts (k):[',IK,']'
      WRITE(11,*)'----------- Time Series Files -------------------',
     +'---------------------'
      WRITE(11,*)'File : ',FILENAME1
      WRITE(11,*)'----------- Results ----------------------------',
     +'---------------------'
      WRITE(11,*)'Forward validation results are in :'
      WRITE(11,*)'C:\FV1.TXT'
      WRITE(11,*)'------------------------------------------------',
     +'---------------------'
      WRITE(11,*)'----------- Grouping of Eigenvectors supplied ---',
     +'---------------------'
      WRITE(11,*)'      Group',' Eigenvectors','      Index 1',
     +'      Index 2'
      DO I=1,IGROUP
       WRITE(11,*)I,' ',IC(I,1),(' ',IC(I,J+1),J=1,IC(I,1))
      ENDDO
      WRITE(11,*)'------------------------------------------------',
     +'---------------------'
      WRITE(11,*)'-------- Sequential Groupings used during FV ----',
     +'---------------------'
C Construct a string that containts the "model" used
      WRITE(11,*)'Model',' Eigenvectors Used ("model")'
      DO I=1,IGROUP
```

```
      STRING1=''
      ICOUNT=1
      STRING1(ICOUNT:ICOUNT)='('
      ICOUNT=ICOUNT+1
       DO K=1,I
        STRING1(ICOUNT:ICOUNT)='['
        ICOUNT=ICOUNT+1
        DO J=1,IC(K,1)
         WRITE(CH3,'(I3)')INT(IC(K,J+1))
         STRING1(ICOUNT:ICOUNT+2)=CH3(1:3)
         ICOUNT=ICOUNT+3
         IF ((IC(K,1).EQ.2).AND.(J.EQ.1)) THEN
          STRING1(ICOUNT:ICOUNT)=','
          ICOUNT=ICOUNT+1
         ENDIF
        ENDDO
        STRING1(ICOUNT:ICOUNT)=']'
        ICOUNT=ICOUNT+1
       ENDDO
       STRING1(ICOUNT:ICOUNT)=')'
       WRITE(11,'(I6,A1,A200)')I,' ',STRING1
      ENDDO
      WRITE(11,*)'------------------------------------------------',
     +'---------------------'

C Count the maximum number of eigenvectors used
      IR1=0
      DO I=1,IGROUP
       DO J=1,IC(I,1)
        IR1=IR1+1
       ENDDO
      ENDDO

C Open OUTPUT file
      OPEN(UNIT=2,FILE='C:\FV1.TXT')

C FORWARD VALIDATION STARTS HERE
C Loop over the window length
      DO 103 ITAU=IR1+IK,ITAU1
C Loop over the sequential models
       DO 102 IR=1,IGROUP
C Set up the vector of index values of eigenvectors to use
C Count the number of eigenvectors in the sequential group
        ICOUNTR=0
        DO I=1,IR
         DO J=1,IC(I,1)
          ICOUNTR=ICOUNTR+1
          IPC(ICOUNTR)=IC(I,J+1)
         ENDDO
        ENDDO
C Set position in original series to start FV at
        IN3=IN1-IK+1
        IT2=IN3
        IN4=IN3
C Initialise MSEMAT matrix for use
        MSE=0.0D0
        MSEMAT(ITAU,IR)=0.0D0

        DO 101 I1=1,IN2
         WRITE(*,'(1X,A3,I4,A6,I4,A8,I4)')'I1=',I1,' ITAU=',ITAU,
     +   ' IGROUP=',IR
C The following routine returns a signal with its shift vector and
C projection matrix. The series is (0) the original (1) the approximate
C (2) the Cadzow-signal series
```

```
         CALL SIGNALSERIES(F,INOBS,IN3,ITAU,ICOUNTR,SIGNAL,XMEAN,PV,
     &   ITERATE,ISERIES)

C Create projection matrix of orthogonal parallel linear subspace Lr
         DO I=1,ITAU
          DO J=I,ITAU
           IF (I.EQ.J) THEN
            P(I,J)=1.0D0-PV(I,J)
           ELSE
            P(I,J)=0.0D0-PV(I,J)
           ENDIF
           P(J,I)=P(I,J)
          ENDDO
         ENDDO

C Partition projection matrix P into P1 and P2
         DO I=1,ITAU
          DO J=1,ITAU
           IF (J.LE.(ITAU-IK)) THEN
            P1(I,J)=P(I,J)
           ENDIF
           IF (J.GE.(ITAU-IK+1)) THEN
            P2(I,J-(ITAU-IK))=P(I,J)
           ENDIF
          ENDDO
         ENDDO

C Create P2'P2 and invert the result
         DO I=1,IK
          DO J=1,IK
           P2TP2(I,J)=0.0D0
           DO IJ=1,ITAU
            P2TP2(I,J)=P2TP2(I,J)+P2(IJ,I)*P2(IJ,J)
           ENDDO
           P2TP2(J,I)=P2TP2(I,J)
          ENDDO
         ENDDO

C Set up inverse of P2TP2
         CALL DLINDS(IK,P2TP2,200,P2TP2INV,200)

C Create P2'P1 and invert the result
         DO I=1,IK
          DO J=1,(ITAU-IK)
           P2TP1(I,J)=0.0D0
           DO IJ=1,ITAU
            P2TP1(I,J)=P2TP1(I,J)+P2(IJ,I)*P1(IJ,J)
           ENDDO
          ENDDO
         ENDDO

C Create inv(P2'P2)P2TP1
         DO I=1,IK
          DO J=1,(ITAU-IK)
           P2TP2IP2TP1(I,J)=0.0D0
           DO IJ=1,IK
            P2TP2IP2TP1(I,J)=P2TP2IP2TP1(I,J)+P2TP2INV(I,IJ)*P2TP1(IJ,J)
           ENDDO
          ENDDO
         ENDDO

C Create k-step-ahead forecasts
         ICOUNT3=0
         DO II=1,ITAU-IK
          ICOUNT3=ICOUNT3+1
          VEC1(ICOUNT3)=SIGNAL(IN3-ITAU+IK+II)-
     +    XMEAN(ICOUNT3)
         ENDDO

C The k-period-ahead forecasts are generated here
C ALL the k-period-ahead forecasts are used to construct the MSE with
         DO I=1,IK
          FHAT(I)=0.0D0
          DO J=1,(ITAU-IK)
           FHAT(I)=FHAT(I)+P2TP2IP2TP1(I,J)*VEC1(J)
          ENDDO
          FHAT(I)=XMEAN(ITAU-IK+I)-FHAT(I)
C Calculate MSE over validation series over all the k-period-ahead forecasts
          IF (ITRANS.EQ.0) THEN
           MSEMAT(ITAU,IR)=MSEMAT(ITAU,IR)+((F(IN4+I)-FHAT(I))**2)/
     +     DFLOAT(IN2*IK)
          ENDIF
          IF (ITRANS.EQ.1) THEN
           MSEMAT(ITAU,IR)=MSEMAT(ITAU,IR)+((F(IN4+I)**2-
     +     FHAT(I)**2)**2)/DFLOAT(IN2*IK)
          ENDIF
          IF (ITRANS.EQ.2) THEN
           MSEMAT(ITAU,IR)=MSEMAT(ITAU,IR)+((DEXP(F(IN4+I))-
     +     DEXP(FHAT(I)))**2)/DFLOAT(IN2*IK)
          ENDIF
         ENDDO
C The training series is incremented by a single observation
         IN3=IN3+1
         IN4=IN4+1
  101   CONTINUE
C Write MSE to OUTPUT files
         WRITE(2,*)ITAU,' ',IR,' ',MSEMAT(ITAU,IR)
  102  CONTINUE
C Increase training series length by 1
  103 CONTINUE

C Calculate mean(MSE) and var(MSE). First initialise the SORT matrix
      DO I=1,IGROUP
       DO J=1,4
        SORTMSE(I,J)=0.0D0
       ENDDO
      ENDDO

      DO I=1,IGROUP
C Store number of leading eigenvectors in column [1] of SORTMSE matrix
       SORTMSE(I,1)=I
C Calculate mean(MSE) and store in column [2] of SORTMSE matrix
       DO J=IR1+IK,ITAU1
        SORTMSE(I,2)=SORTMSE(I,2)+MSEMAT(J,I)/DFLOAT(ITAU1-IR1-IK+1)
       ENDDO
C Calculate var(MSE) and store in column [3] of SORTMSE matrix
       DO J=IR1+IK,ITAU1
        SORTMSE(I,3)=SORTMSE(I,3)+((MSEMAT(J,I)-SORTMSE(I,2))**2)/
     + DFLOAT(ITAU1-IR1-IK)
       ENDDO
C Calculate mean(MSE)^2+var(MSE) and store in column [4] of SORTMSE matrix
       SORTMSE(I,4)=SORTMSE(I,2)**2+SORTMSE(I,3)
      ENDDO

C Sort the results by mean(MSE)^2+var(MSE), i.e. by column [4] of SORTMSE matrix
      DO I=1,IGROUP
       DO J=I,IGROUP
        IF (SORTMSE(J,4).LT.SORTMSE(I,4)) THEN
```

```
         TEMP1=SORTMSE(I,1)
         TEMP2=SORTMSE(I,2)
         TEMP3=SORTMSE(I,3)
         TEMP4=SORTMSE(I,4)
         SORTMSE(I,1)=SORTMSE(J,1)
         SORTMSE(I,2)=SORTMSE(J,2)
         SORTMSE(I,3)=SORTMSE(J,3)
         SORTMSE(I,4)=SORTMSE(J,4)
         SORTMSE(J,1)=TEMP1
         SORTMSE(J,2)=TEMP2
         SORTMSE(J,3)=TEMP3
         SORTMSE(J,4)=TEMP4
        ENDIF
       ENDDO
      ENDDO

C Write sorted results to file
      WRITE(11,*)' '
      WRITE(11,*)'FV results sorted by mean(MSE)^2+var(MSE)'
      WRITE(11,'(3X,A5,7X,A9,22X,A8,23X,A20)')
     +'Model','mean(MSE)','var(MSE)','mean(MSE)^2+var(MSE)'
      DO I=1,IGROUP
       WRITE(11,'(I4,3(A1,G30.15))')INT(SORTMSE(I,1)),' ',SORTMSE(I,2),
     + ' ',SORTMSE(I,3),' ',SORTMSE(I,4)
      ENDDO

C Sort the results by var(MSE), i.e. by column [3] of SORTMSE matrix
      DO I=1,IGROUP
       DO J=I,IGROUP
        IF (SORTMSE(J,3).LT.SORTMSE(I,3)) THEN
         TEMP1=SORTMSE(I,1)
         TEMP2=SORTMSE(I,2)
         TEMP3=SORTMSE(I,3)
         TEMP4=SORTMSE(I,4)
         SORTMSE(I,1)=SORTMSE(J,1)
         SORTMSE(I,2)=SORTMSE(J,2)
         SORTMSE(I,3)=SORTMSE(J,3)
         SORTMSE(I,4)=SORTMSE(J,4)
         SORTMSE(J,1)=TEMP1
         SORTMSE(J,2)=TEMP2
         SORTMSE(J,3)=TEMP3
         SORTMSE(J,4)=TEMP4
        ENDIF
       ENDDO
      ENDDO

C Write sorted results to file
      WRITE(11,*)' '
      WRITE(11,*)'FV results sorted by var(MSE)'
      WRITE(11,'(3X,A5,7X,A9,22X,A8,23X,A20)')
     +'Model','mean(MSE)','var(MSE)','mean(MSE)^2+var(MSE)'
      DO I=1,IGROUP
       WRITE(11,'(I4,3(A1,G30.15))')INT(SORTMSE(I,1)),' ',SORTMSE(I,2),
     + ' ',SORTMSE(I,3),' ',SORTMSE(I,4)
      ENDDO

C Close the OUTPUT file
      CLOSE(2)

C Write the time and date that analysis ended to file
      WRITE(11,*)'--------------------------------------------------',
     +'---------------------'
      CALL TIMDY (IHOUR, IMIN, ISEC)
      CALL TDATE (IDAY, IMONTH, IYEAR)
```

```
      WRITE(11,10)'Date analysis ended         : [ ',IDAY,':',IMONTH,':',
     +IYEAR,'  ]'
      WRITE(11,20)'Time analysis ended         : [ ',IHOUR,':',IMIN,':',
     +ISEC,'    ]'
      WRITE(11,*)'--------------------------------------------------',
     +'---------------------'
      CLOSE(11)
      WRITE(*,*)'Comments and sorted results are in file C:\OUT.TXT'
      WRITE(*,*)'Pivot table data for EXCEL is in file C:\FV1.TXT'
      END

C ********************************************************************
C                 SIGRECONS [J. DE KLERK, 2002]
C ********************************************************************
C This subroutine gathers information for signal reconstruction
C ********************************************************************
      SUBROUTINE SIGRECONS(ICHOICE2)
C ********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      CHARACTER*60 FILENAME1,FILENAME2
      CHARACTER*4 TRANSFORM(3)
      DIMENSION F(IMAXOBS),SIGNAL(IMAXOBS)
      DIMENSION X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION XXT(IMAXTAU,IMAXTAU),V(IMAXTAU,IMAXTAU),EVAL(IMAXTAU)
      DIMENSION PV(IMAXTAU,IMAXTAU),XTILDE(IMAXTAU,IMAXCOLS)
      DIMENSION VEC1(IMAXTAU),VEC2(IMAXTAU),XHAT(IMAXTAU,IMAXCOLS)
      DIMENSION DIAGMEAN(IMAXOBS),IDIAG(IMAXOBS)
      DATA TRANSFORM/'NONE','SQRT','NLOG'/

C Get INPUT and OUTPUT filenames from user
      WRITE(*,1)'INPUT filename  :                                   '
      READ(*,*)FILENAME1
      WRITE(*,1)'OUTPUT filename :                                   '
      READ(*,*)FILENAME2
      WRITE(*,1)'Apply 1. NO 2. SQRT 3. LN transformation :          '
      READ(*,*)ITRANS
      OPEN(UNIT=1,FILE=FILENAME1,STATUS='OLD')
      OPEN(UNIT=2,FILE=FILENAME2)
    1 FORMAT(A51)

C Read Time Series into vector F while counting the number of OBS
C Also store time series as iteration=1 signal series
      INOBS=1
    3 READ(1,*,END=4)F(INOBS)
       IF (ITRANS.EQ.2) THEN
        F(INOBS)=DSQRT(F(INOBS))
       ENDIF
       IF (ITRANS.EQ.3) THEN
        F(INOBS)=DLOG(F(INOBS))
       ENDIF
       SIGNAL(INOBS)=F(INOBS)
       INOBS=INOBS+1
      GOTO 3
    4 INOBS=INOBS-1
      CLOSE(1)

C Get [ITAU] and [IR] from user
```

```
      WRITE(*,*)'Length of series is [',INOBS,']'
      WRITE(*,1)'Length of time series to smooth              :          '
      READ(*,*)INOBS
      WRITE(*,1)'Enter a value for [tau]     :                    '
      READ(*,*)ITAU
      WRITE(*,1)'Enter a value for [r]       :                    '
      READ(*,*)IR

C Check user supplied tau
      IF (ITAU.GT.INT((INOBS+1)/2)) THEN
       ITAU=INT((INOBS+1)/2)
      ENDIF
C Check user supplied r
      IF (IR.GE.ITAU) THEN
       IR=ITAU-1
      ENDIF

C Write information to OUTPUT FILE
      WRITE(2,*)'--------------------------------------------------'
      WRITE(2,*)'                 Shongololo-SSA                   '
      WRITE(2,*)'                  developed by                    '
      WRITE(2,*)'                                                  '
      WRITE(2,*)'                  J. de Klerk (2002)              '
      WRITE(2,*)'  Department of Statistics and Actuarial Science  '
      WRITE(2,*)'              Stellenbosch University             '
      WRITE(2,*)'              Republic of South Africa            '
      WRITE(2,*)'               email: jdk2@sun.ac.za              '
      WRITE(2,*)'--------------------------------------------------'
      WRITE(2,*)' The program uses forecasting algorithms that form '
      WRITE(2,*)' part of the developer''s PhD thesis, titled:-    '
      WRITE(2,*)'    Time Series Forecasting and Model Selection   '
      WRITE(2,*)'                        in                        '
      WRITE(2,*)'             Singular Spectrum Analysis           '
      WRITE(2,*)'--------------------------------------------------'
      WRITE(2,*)'----------- Time Series Information -------------',
     +'---------------------'
      CALL TIMDY (IHOUR, IMIN, ISEC)
      CALL TDATE (IDAY, IMONTH, IYEAR)
      WRITE(2,10)'Date of analysis              : [ ',IDAY,':',IMONTH,':',
     +IYEAR,'  ]'
      WRITE(2,20)'Time of analysis              : [ ',IHOUR,':',IMIN,':',
     +ISEC,'    ]'
      WRITE(2,*)'Running subroutine            : [  SIGRECONS   ]'
   10 FORMAT (1X,A31,I2,A1,I2,A1,I4,A4)
   20 FORMAT (1X,A31,I2,A1,I2,A1,I2,A6)
      WRITE(2,11)'Input Filename                : [ ',FILENAME1,' ]'
   11 FORMAT(1X,A31,1X,A60,A2)
      WRITE(2,*)'Time Series length_N          : [ ',INOBS,' ]'
      WRITE(2,*)'Value of tau                  : [ ',ITAU,' ]'
      WRITE(2,*)'Value of r                    : [ ',IR,' ]'
      IF (ICHOICE2.EQ.1) THEN
       WRITE(2,*)'Type of signal                : [ Approx ]'
      ENDIF
      IF (ICHOICE2.EQ.2) THEN
       WRITE(2,*)'Type of signal                : [ Cadzow ]'
      ENDIF

      WRITE(2,*)'----------- Notes --------------------------------',
     +'---------------------'
      WRITE(2,*)'Transformation applied to series     : ',
     +TRANSFORM(ITRANS)
      WRITE(2,*)'--------------------------------------------------',
     +'---------------------'
```

```
C Call the actual subroutine that performs signal reconstruction
      CALL SIGNALSERIES2(F,INOBS,INOBS,ITAU,IR,SIGNAL,XMEAN,PV,
     +ITERATE,ICHOICE2)

      WRITE(2,65)'CONVERGENCE IN ',ITERATE,' ITERATIONS'
   65 FORMAT(1X,A15,G10.0,A11)
      WRITE(2,*)'--------------------------------------------------',
     +'---------------------'

      WRITE(2,66)'ORIGINAL SERIES','SIGNAL SERIES'
   66 FORMAT(18X,A15,26X,A13)

C Write results to file and back-transform if necessary
      DO I=1,INOBS
       IF (ITRANS.EQ.2) THEN
        F(I)=F(I)**2
        SIGNAL(I)=SIGNAL(I)**2
       ENDIF
       IF (ITRANS.EQ.3) THEN
        F(I)=DEXP(F(I))
        SIGNAL(I)=DEXP(SIGNAL(I))
       ENDIF
       WRITE(2,67)F(I),SIGNAL(I)
      ENDDO
   67 FORMAT(2(G40.18,2X))
      IF (ITRANS.NE.1) THEN
       WRITE(2,*)'--------------------------------------------------',
     + '---------------------'
       WRITE(2,*)'A ',TRANSFORM(ITRANS),' transformation was applied',
     + ' before smoothing the tıme series.'
       WRITE(2,*)'Results above are the back-transformed series.'
      ENDIF
C Calculate how well smoothed series fits the time series
      CALL MEASURES(INOBS,F,SIGNAL,MAD,MAPE,SSE,MSE)
      WRITE(2,*)'----------- Measures of Fitting Accuracy ----',
     +'------------------------'
      WRITE(2,'(A6,22X,I5)')' N   =',INOBS
      WRITE(2,'(A6,1X,G30.15)')' SSE =',SSE
      WRITE(2,'(A6,1X,G30.15)')' MSE =',MSE
      WRITE(2,'(A6,1X,G30.15)')' MAD =',MAD
      WRITE(2,'(A6,1X,G30.15)')' MAPE=',MAPE
      WRITE(2,*)'--------------------------------------------------',
     + '---------------------'
      CLOSE(2)
      END


C ******************************************************************
C                  FORECASTING [J. DE KLERK, 2002]
C ******************************************************************
      SUBROUTINE FORECASTING
C ******************************************************************
C This subroutine is used to produce forecasts
C The subroutine gathers the required information and then calls the
C the correct subroutines
C ******************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)
      DIMENSION F(IMAXOBS),SIGNAL(IMAXOBS),FHAT(200),IEVEC(40)
      CHARACTER*60 FILENAME1,FILENAME2,FILENAME3
      CHARACTER*51 ITECH(3),ISERIES(3)
      DIMENSION X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU),MEASURES(5)
      DIMENSION PV(IMAXTAU,IMAXTAU),EXPLAINED(IMAXTAU)
      CHARACTER*4 TRANS(3)

C Store character labels that will be used later
      TRANS(1)='None'
      TRANS(2)='Log '
      TRANS(3)='Sqrt'
      ITECH(1)='1=Recurrent one-period-ahead forecasts             '
      ITECH(2)='2=Recurrent one-period-ahead forecasts re-estimated'
      ITECH(3)='3=Joint-horizon k-period-ahead forecasts           '
      ISERIES(1)='1=Original series                                  '
      ISERIES(2)='2=Approximate series                               '
      ISERIES(3)='3=Cadzow reconstructed series                      '

    1 FORMAT(A51)
      WRITE(*,1)'---------------------------------------------------'
      WRITE(*,1)'------------ FORECASTING OPTION -------------------'

C Get FILENAME of file that contains FILENAMES from user
      WRITE(*,1)'FILENAME containing time series                    '
      READ(*,*)FILENAME1

C Get OUTPUT filename
      WRITE(*,1)'OUTPUT filename                                    '
      READ(*,*)FILENAME2

C Get TRANSFORMATION to use
      WRITE(*,1)'TRANSFORMATION to use                              '
      WRITE(*,1)'1=None                                             '
      WRITE(*,1)'2=Natural log                                      '
      WRITE(*,1)'3=Square root                                      '
      READ(*,*)ITRANS

C Read time series into vector F, counting the number of observations
C Also apply the required transformation
      OPEN(UNIT=1,FILE=FILENAME1)
      INOBS=1
    3 READ(1,*,END=4)F(INOBS)
       IF (ITRANS.EQ.2) THEN
        F(INOBS)=LOG(F(INOBS))
       ENDIF
       IF (ITRANS.EQ.3) THEN
        F(INOBS)=SQRT(F(INOBS))
       ENDIF
       INOBS=INOBS+1
      GOTO 3
    4 INOBS=INOBS-1
      CLOSE(1)
      WRITE(*,1)'---------------------------------------------------'

C Write time series length to screen
      WRITE(*,'(A22,I4)')'Time series length N= ',INOBS

C Get index that forecasts start at
      WRITE(*,1)'---------------------------------------------------'
      WRITE(*,1)'Index value at which forecasts start at            '
      READ(*,*)ISTART
      IF (ISTART.GT.(INOBS+1)) THEN
       ISTART=(INOBS+1)
      ENDIF
```

```
C Get number of forecasts to generate
      WRITE(*,1)'Number of forecasts to generate (k)                '
      READ(*,*)IK
      WRITE(*,1)'-- SSA Parameters ---------------------------------'

C Use centring or non-centring
      WRITE(*,1)'Use centring 1=YES 0=NO                            '
      READ(*,*)ICENT

C Get window length
      WRITE(*,1)'Window length to use (tau)                         '
      READ(*,*)ITAU

C Check supplied value for window length
      IF (ITAU.GT.(MIN(INT((INOBS)/2),INT((ISTART)/2)))) THEN
       ITAU=MIN(INT((INOBS)/2),INT((ISTART)/2))
      ENDIF

C Get number of eigenvectors to use
      WRITE(*,1)'Number of eigenvectors to use (r)                  '
      READ(*,*)IR
      DO I=1,IR
       WRITE(*,'(A21,I4,A1)')'INDEX OF EIGENVECTOR[',I,']'
       READ(*,*)IEVEC(I)
      ENDDO

C Get series to use during forecasting
      WRITE(*,1)
      WRITE(*,1)'Series to use during forecasting                   '
      WRITE(*,1)ISERIES(1)
      WRITE(*,1)ISERIES(2)
      WRITE(*,1)ISERIES(3)
      READ(*,*)ICH1

C Get forecasting technique to use
      WRITE(*,1)
      WRITE(*,1)'Forecasting technique to use                       '
      WRITE(*,1)ITECH(1)
      WRITE(*,1)ITECH(2)
      WRITE(*,1)ITECH(3)
      READ(*,*)ICH2
      WRITE(*,1)'---------------------------------------------------'

C Perform recurrent one-period-ahead forecasting, if requested
      IF (ICH2.EQ.1) THEN
       CALL FORECAST1(F,INOBS,ISTART,ITAU,IR,IEVEC,IK,ICENT,ICH1,
     + FHAT,MAD,SSE,MSE,MAPE,ITRANS)
      ENDIF

C Perform recurrent one-period-ahead forecasting, if requested
C Re-estimation is used
      IF (ICH2.EQ.2) THEN
       CALL FORECAST2(F,INOBS,ISTART,ITAU,IR,IEVEC,IK,ICENT,ICH1,
     + FHAT,MAD,SSE,MSE,MAPE,ITRANS)
      ENDIF

C Perform joint-horizon k-period-ahead forecasting, if requested
      IF (ICH2.EQ.3) THEN
       CALL FORECAST3(F,INOBS,ISTART,ITAU,IR,IEVEC,IK,ICENT,ICH1,
    +  FHAT,MAD,SSE,MSE,MAPE,ITRANS)
       ENDIF

C Write information to OUTPUT FILE
      OPEN(UNIT=2,FILE=FILENAME2)
```

```
C Open file 'C:\SHONG.DAT' which contains % variation explained
      OPEN(UNIT=4,FILE='C:\SHONG.DAT')

C Read % variation explained and calculate total sum % variation
      SUM1=0
      DO I=1,IR
       READ(4,*)EXPLAINED(I)
       SUM1=SUM1+EXPLAINED(I)
      ENDDO
      CLOSE(4)

C Write results to file
      WRITE(2,*)'-------------------------------------------------'
      WRITE(2,*)'               Shongololo-SSA                    '
      WRITE(2,*)'                developed by                     '
      WRITE(2,*)'                                                 '
      WRITE(2,*)'                 J. de Klerk (2002)              '
      WRITE(2,*)' Department of Statistics and Actuarial Science  '
      WRITE(2,*)'              Stellenbosch University            '
      WRITE(2,*)'              Republic of South Africa           '
      WRITE(2,*)'               email: jdk2@sun.ac.za             '
      WRITE(2,*)'-------------------------------------------------'
      WRITE(2,*)' The program uses forecasting algorithms that form '
      WRITE(2,*)' part of the developer''s PhD thesis, titled:-   '
      WRITE(2,*)'    Time Series Forecasting and Model Selection  '
      WRITE(2,*)'                     in                          '
      WRITE(2,*)'           Singular Spectrum Analysis            '
      WRITE(2,*)'-------------------------------------------------'

      WRITE(2,*)'----------- Time Series Information -------------',
     + '---------------------'
      CALL TIMDY (IHOUR, IMIN, ISEC)
      CALL TDATE (IDAY, IMONTH, IYEAR)
      WRITE(2,10)'Date of analysis          : [ ',IDAY,':',IMONTH,':',
     + IYEAR,'  ]'
      WRITE(2,20)'Time of analysis          : [ ',IHOUR,':',IMIN,':',
     + ISEC,'    ]'
      WRITE(2,*)'Running subroutine       : [ FORECASTING  ]'
   10 FORMAT (1X,A31,I2,A1,I2,A1,I4,A4)
   20 FORMAT (1X,A31,I2,A1,I2,A1,I2,A6)
      WRITE(2,11)'Input Filename            : [ ',FILENAME1,' ]'
   11 FORMAT(1X,A31,1X,A60,A2)
      WRITE(2,*)'Time Series length_N     : [ ',INOBS,' ]'
      WRITE(2,*)'Window Length (tau)      : [ ',ITAU,' ]'
      WRITE(2,*)'Transformation used      : [ ',TRANS(ITRANS),' ]'
      IF (ICENT.EQ.1) THEN
       WRITE(2,*)'Centring used             : [          Yes ]'
      ELSE
       WRITE(2,*)'Centring used             : [           No ]'
      ENDIF
      IF (ICH1.EQ.1) THEN
       WRITE(2,*)'Forecasted with series    : [     original ]'
      ENDIF
      IF (ICH1.EQ.2) THEN
       WRITE(2,*)'Forecasted with series    : [ reconstucted ]'
      ENDIF
      IF (ICH1.EQ.3) THEN
       WRITE(2,*)'Forecasted with series    : [Cadzow signal ]'
      ENDIF
      IF (ICH2.EQ.1) THEN
       WRITE(2,*)'Forecasting technique used : [ recurrent 1- ]'
      ENDIF
      IF (ICH2.EQ.2) THEN
       WRITE(2,*)'Forecasting technique used : [ recurrent re ]'
      ENDIF
      IF (ICH2.EQ.3) THEN
       WRITE(2,*)'Forecasting technique used : [joint horizon ]'
      ENDIF
      WRITE(2,*)'Forecasts starts at        : [ ',ISTART,' ]'
      WRITE(2,*)'Number of forecasts        : [ ',IK,' ]'
      WRITE(2,*)'Number of eigenvectors used: [ ',IR,' ]'
      WRITE(2,*)' '
      WRITE(2,*)'List of eigenvectors used  :'

C Write eigenvectors and % variation explained to file
      DO I=1,IR
       WRITE(2,'(1X,A13,I4,A2,7X,A4,I4,A4,F8.4,A3)')
     + 'Eigenvector[ ',I,' ]',' : [',IEVEC(I),' ][ ',EXPLAINED(I)*100,
     + '% ]'
      ENDDO

C Write total sum % variation explained
      WRITE(2,'(37X,A2,F8.4,A3)')'[ ',(SUM1*100),'% ]'
      WRITE(2,*)' '
      WRITE(2,*)' '
      WRITE(2,*)'FORECAST RESULTS'
      WRITE(2,*)'-------------------------------------------------',
     + '--------------'

C Write Forecasts and Measure of forecasting accuracy to file
      WRITE(2,'(1X,A4,16X,A11,24X,A8,22X,A10,6X,A11)')'Time',
     + 'Time Series','Forecast','ABS(Error)','ABS % Error'
      WRITE(2,*)'-------------------------------------------------',
     + '--------------'

      DO I=1,IK
       IF (ITRANS.EQ.1) THEN
        IF ((ISTART+I-1).LE.INOBS) THEN
         WRITE(2,'(I4,3(2X,G30.15),2X,F10.7,A1)')(ISTART+I-1),
     +   F(ISTART+I-1),FHAT(I),DABS(F(ISTART+I-1)-FHAT(I)),
     +   DABS(F(ISTART+I-1)-FHAT(I))/(F(ISTART+I-1))*100,'%'
         ELSE
          WRITE(2,'(I4,34X,G30.15)')(ISTART+I-1),FHAT(I)
        ENDIF
       ENDIF
       IF (ITRANS.EQ.2) THEN
        IF ((ISTART+I-1).LE.INOBS) THEN
             WRITE(2,'(I4,3(2X,G30.15),2X,F10.7,A1)')(ISTART+I-1),
     +   DEXP(F(ISTART+I-1)),DEXP(FHAT(I)),DABS(DEXP(F(ISTART+I-1))
     +   -DEXP(FHAT(I))),DABS(DEXP(F(ISTART+I-1))-DEXP(FHAT(I)))/
     +   (DEXP(F(ISTART+I-1)))*100,'%'
        ELSE
         WRITE(2,'(I4,34X,G30.15)')(ISTART+I-1),DEXP(FHAT(I))
        ENDIF
       ENDIF
       IF (ITRANS.EQ.3) THEN
        IF ((ISTART+I-1).LE.INOBS) THEN
             WRITE(2,'(I4,3(2X,G30.15),2X,F10.7,A1)')(ISTART+I-1),
     +   DEXP(F(ISTART+I-1)),(FHAT(I)**2),DABS((F(ISTART+I-1)**2)
     +   -(FHAT(I)**2)),DABS((F(ISTART+I-1)**2)-(FHAT(I)**2))/
     +   ((F(ISTART+I-1)**2))*100,'%'
        ELSE
         WRITE(2,'(I4,34X,G30.15)')(ISTART+I-1),(FHAT(I)**2)
        ENDIF
       ENDIF
      ENDDO
      WRITE(2,*)'------------------------------------------------',
     + '---------------'
      IF ((ISTART-1).LT.INOBS) THEN
       WRITE(2,*)'MEASURES OF FORECASTING ACCURACY'
       WRITE(2,*)'--------------------------------'
       WRITE(2,'(A6,22X,I5)')' N    =',(INOBS-ISTART+1)
       WRITE(2,'(A6,1X,G30.15)')' MAD =',MAD
       WRITE(2,'(A6,1X,G30.15)')' SSE =',SSE
       WRITE(2,'(A6,1X,G30.15)')' MSE =',MSE
       WRITE(2,'(A6,16X,F10.7,A1)')' MAPE=',MAPE,'%'
       WRITE(2,*)'--------------------------------'
      ENDIF
      CLOSE(2)
      END

C ******************************************************************
C                   BATCH [J. DE KLERK, 2002]
C ******************************************************************
c This subroutine obtains all the necessary input values to perform
C batch forward validation. The routine the calls BATCHFVAL
C ******************************************************************
      SUBROUTINE BATCH
C ******************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      CHARACTER*60 FILENAME1,FILENAME2,FILENAME3
      DIMENSION F(IMAXOBS)

    1 FORMAT(A51)
      WRITE(*,1)'--------------------------------------------------'
      WRITE(*,1)'This program performs Batch Forward Validation.   '
      WRITE(*,1)'Rolling k-period-ahead forecasts are used.        '
      WRITE(*,1)'The technique uses centring of trajectory matrix  '
      WRITE(*,1)'--------------------------------------------------'


C Get file with time series filenames from user
      WRITE(*,'(A29)')'Filename containg filenames: '
      READ(*,*)FILENAME1

C Get OUTPUT filename
      WRITE(*,'(A29)')'OUTPUT filename             : '
      READ(*,*)FILENAME3

C Open the OUTPUT filename
      OPEN(UNIT=3,FILE=FILENAME3)

C Write output to OUPUT file
      WRITE(3,*)'--------------------------------------------------'
      WRITE(3,*)'                 Shongololo-SSA                   '
      WRITE(3,*)'                  developed by                    '
      WRITE(3,*)'                                                  '
      WRITE(3,*)'                   J. de Klerk (2002)             '
      WRITE(3,*)'  Department of Statistics and Actuarial Science  '
      WRITE(3,*)'                Stellenbosch University           '
      WRITE(3,*)'                Republic of South Africa          '
      WRITE(3,*)'                 email: jdk2@sun.ac.za            '
      WRITE(3,*)'--------------------------------------------------'
```

```
      WRITE(3,*)' The program uses forecasting algorithms that form '
      WRITE(3,*)' part of the developer''s PhD thesis, titled:-     '
      WRITE(3,*)'    Time Series Forecasting and Model Selection   '
      WRITE(3,*)'                        in                        '
      WRITE(3,*)'             Singular Spectrum Analysis           '
      WRITE(3,*)'--------------------------------------------------'
      WRITE(3,*)'-------------------------------------------------',
     +'---------------------'
      WRITE(3,*)' Normal Batch Forward Validation Results         '
      WRITE(3,*)'-------------------------------------------------',
     +'---------------------'
      WRITE(3,*)'----------- Date of Analysis and Program Used ---',
     +'---------------------'
      CALL TIMDY (IHOUR, IMIN, ISEC)
      CALL TDATE (IDAY, IMONTH, IYEAR)
   10 FORMAT (1X,A31,I2,A1,I2,A1,I4,A4)
   22 FORMAT (1X,A31,I2,A1,I2,A1,I2,A6)
      WRITE(3,10)'Date of analysis started   : [ ',IDAY,':',IMONTH,':',
     +IYEAR,'  ]'
      WRITE(3,22)'Time of analysis started   : [ ',IHOUR,':',IMIN,':',
     +ISEC,'    ]'
      WRITE(3,*)'Running subroutine         : [  BATCHFVAL   ]'
      WRITE(3,*)'-------------------------------------------------',
     +'---------------------'

      WRITE(3,'(A2,24X,A26,24X,A2,2X,A2,6X,A33,103X,A2)')
     +'|-','FORWARD VALIDATION RESULTS','-|',
     +'|-','OUT-OF-SAMPLE FORECASTING RESULTS','-|'
      WRITE(3,'(A20,15X,A1,5X,A2,5X,A2,1X,A6,3X,A4,6X,A1,6X,A1,2X,A5,5X,
     +A2,4X,A3,6X,A1,24X,A3,28X,A3,28X,A3,27X,A4)')
     +'Time Series Filename','N','N1','N2','MAXtau','MAXr','k','r',
     +'N1+N2','N3','tau','r','MAD','SSE','MSE','MAPE'

C Get series to use during FV from user
      WRITE(*,'(A43)')'User series 0=Original, 1=Approx, 2=Cadzow:'
      READ(*,*)ISERIES

C Get k-step-ahead forecasts from user
      WRITE(*,'(A27)')'k-period-ahead forecasts : '
      READ(*,*)IK

C Open file with filenames and read filenames
      OPEN(UNIT=1,FILE=FILENAME1)
   11 READ(1,*,END=55)FILENAME2

C Read time series into vector F, counting the number of observations
       OPEN(UNIT=2,FILE=FILENAME2)
       INOBS=1
    2  READ(2,*,END=3)F(INOBS)
        INOBS=INOBS+1
       GOTO 2
    3  INOBS=INOBS-1
       CLOSE(2)

C Set the validation series length. A very minimum length of 3 is imposed!
       IF (INT(0.1*INOBS).GT.3) THEN
        IN2=MIN(INT(0.1*INOBS),12)
       ELSE
        IN2=3
       ENDIF

C Set train series length
       IN1=INOBS-2*IN2
```

```
C Set maximum window length to loop over up to maximum of 70
       ITAU1=MIN(INT((IN1+1)/2),70)

C Set maximum number of leading eigenvectors to loop over
C Check that (IMAXTAU1-(IMAXR+IK)+1)>2.
       IF (ITAU1-7.GT.0) THEN
        IR1=MIN(ITAU1-7,13)
       ELSE
        IR1=MIN(ITAU1-2,13)
       ENDIF
C Set k-period ahead forecasting during FV
       IF ((IN1-ITAU1).GE.6) THEN
        IK=MIN(ITAU1-IR1-1,12)
       ELSE
        IK=1
       ENDIF

C Reset the maximum window length to use
       ITAU1=MIN(INT((IN1-IK+2)/2),70)

C Perform Forward Validation
       CALL BATCHFVAL(F,INOBS,IN1,IN2,IK,IR1,ITAU1,ISERIES,MSE1,MSE2,
     + MSE3,ITAU2,IMODEL)

C Set the number of leading eigenvectors to use
       IR=IMODEL

C Set window length to use. Set this to the maximum possible length
       ITAU=INT((IN1+IN2+1)/2)

C Set one-period-ahead forecasting to use
       IK1=1

C Set length of test set. It is same length as validation set. IN2=IN3
       IN3=IN2

C Calculate out-of-sample measures of forecasting accuracy
       CALL FORECAST(F,IN1+IN2,IN3,ITAU,IR,IK1,MAD,SSE,MSE,MAPE)

C Write results to OUTPUT file
   20  FORMAT(A29,11(1X,I6),4(1X,G30.15))
       WRITE(3,20)FILENAME2,INOBS,IN1,IN2,ITAU1,IR1,IK,IR,
     + IN1+IN2,IN3,ITAU,IR,MAD,SSE,MSE,MAPE

      GOTO 11
   55 CLOSE(1)
C Write the time and date that analysis ended to file
      CALL TIMDY (IHOUR, IMIN, ISEC)
      CALL TDATE (IDAY, IMONTH, IYEAR)
      WRITE(3,10)'Date of analysis ended     : [ ',IDAY,':',IMONTH,':',
     +IYEAR,'  ]'
      WRITE(3,22)'Time of analysis ended     : [ ',IHOUR,':',IMIN,':',
     +ISEC,'    ]'
      CLOSE(3)
      END

C *********************************************************************
C                   BATCHFVAL [J. DE KLERK, 2002]
C *********************************************************************
      SUBROUTINE BATCHFVAL(F,INOBS,IN1,IN2,IK,IR1,ITAU1,ISERIES,
     +MSE1,MSE2,MSE3,ITAU2,IMODEL)
C *********************************************************************
C This subroutine performs the actual Batch Forward Validation
C ********************VARIABLES****************************************
C FHAT      : k-step-ahead forecasts
C INOBS     : time series length
C IP        : Number of time series
C SIGNAL    : reconstructed signal series
C ICOLS     : number of columns in matrix X  (sample size)
C X         : trajectory Matrix
C XTILDE    : centred trajectory matrix
C XXT       : scatter matrix of matrix XTILDE
C V         : matrix of eigenvectors of the scatter matrix XXT
C EVAL      : vector of eigenvalues of the scatter matrix XXT
C PV        : projection matrix of parallel linear subspace Vr
C P         : projection matrix of orthogonal parallel linear subspace
C P1        : partitioning of matrix P
C P2        : partitioning of matrix P
C P2TP2     : matrix P2'*P2
C P2TP1     : matrix P2'*P1
C P2TP2INV  : matrix inv(P2'P2)
C XMEAN     : shift vector of r-flat
C ***********INPUT VARIABLES*****************************************
C F          : Vector of time series observations
C INOBS      : Original time series length
C IN1        : Length of training series
C IN2        : Length of validation series
C ITAU1      : Maximum window length to use
C IR1        : Maximum number of leading eigenvectors to use
C ***********OUTPUT VARIABLES****************************************
C IMODEL     : Number of leading eigenvectors to use
C *********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      DIMENSION F(IMAXOBS),SIGNAL(IMAXOBS),XMEAN(IMAXTAU)
      DIMENSION X(IMAXTAU,IMAXCOLS),XTILDE(IMAXTAU,IMAXCOLS)
      DIMENSION V(IMAXTAU,IMAXTAU)
      DIMENSION XXT(IMAXTAU,IMAXTAU),EVAL(IMAXTAU)
      DIMENSION FHAT(IMAXOBS),PV(IMAXTAU,IMAXTAU)
      DIMENSION P1(IMAXTAU,IMAXTAU),P2(IMAXTAU,IMAXTAU)
      DIMENSION P(IMAXTAU,IMAXTAU),P2TP2(200,200)
      DIMENSION P2TP1(IMAXTAU,IMAXTAU),P2TP2INV(200,200)
      DIMENSION P2TP2IP2TP1(IMAXTAU,IMAXTAU)
      DIMENSION VEC1(IMAXTAU)
      DIMENSION MSEMAT(IMAXTAU,IMAXTAU-1),SORTMSE(100,3)

C Set ITRANS=0 not to use a transformation
      ITRANS=0
C FORWARD VALIDATION STARTS HERE
      DO 103 ITAU=IR1+IK,ITAU1
       DO 102 IR=1,MIN(ITAU-IK,ITAU-1,IR1)
C Set position in original series to start at
        IN3=IN1-IK+1
        IT2=IN3
        IN4=IN3
C Initialize vector for use
        MSE=0.0D0
        MSEMAT(ITAU,IR)=0.0D0

        DO 101 I1=1,IN2
C The routine that is called does NOT use reconstructed Cadzow-signal
C The convergence criteria has been set to perform no iterations
```

```
C The subroutine is used to return the shift vector and projection matrix
         CALL SIGNALSERIES2(F,INOBS,IN3,ITAU,IR,SIGNAL,XMEAN,PV,
     &   ITERATE,ISERIES)

C Create projection matrix of orthogonal parallel linear subspace Vr
         DO I=1,ITAU
          DO J=I,ITAU
           IF (I.EQ.J) THEN
            P(I,J)=1.0D0-PV(I,J)
           ELSE
            P(I,J)=0.0D0-PV(I,J)
           ENDIF
           P(J,I)=P(I,J)
          ENDDO
         ENDDO

C Partition projection matrix P into P1 and P2
         DO I=1,ITAU
          DO J=1,ITAU
           IF (J.LE.(ITAU-IK)) THEN
            P1(I,J)=P(I,J)
           ENDIF
           IF (J.GE.(ITAU-IK+1)) THEN
            P2(I,J-(ITAU-IK))=P(I,J)
           ENDIF
          ENDDO
         ENDDO

C Create P2'P2 and invert the result
         DO I=1,IK
          DO J=1,IK
           P2TP2(I,J)=0.0D0
           DO IJ=1,ITAU
            P2TP2(I,J)=P2TP2(I,J)+P2(IJ,I)*P2(IJ,J)
           ENDDO
           P2TP2(J,I)=P2TP2(I,J)
          ENDDO
         ENDDO

C Set up inverse of P2TP2
         CALL DLINDS(IK,P2TP2,200,P2TP2INV,200)

C Create P2'P1 and invert the result
         DO I=1,IK
          DO J=1,(ITAU-IK)
           P2TP1(I,J)=0.0D0
           DO IJ=1,ITAU
            P2TP1(I,J)=P2TP1(I,J)+P2(IJ,I)*P1(IJ,J)
           ENDDO
          ENDDO
         ENDDO

C Create inv(P2'P2)P2TP1
         DO I=1,IK
          DO J=1,(ITAU-IK)
           P2TP2IP2TP1(I,J)=0.0D0
           DO IJ=1,IK
            P2TP2IP2TP1(I,J)=P2TP2IP2TP1(I,J)+P2TP2INV(I,IJ)*P2TP1(IJ,J)
           ENDDO
          ENDDO
         ENDDO

C Create k-step-ahead forecasts for the multiple time series
         ICOUNT3=0
         DO II=1,ITAU-IK
          ICOUNT3=ICOUNT3+1
          VEC1(ICOUNT3)=SIGNAL(IN3-ITAU+IK+II)-
     +    XMEAN(ICOUNT3)
         ENDDO

C The k-period-ahead forecasts are generated here
C ALL the k-period-ahead forecasts are used to construct the MSE with
         DO I=1,IK
          FHAT(I)=0.0D0
          DO J=1,(ITAU-IK)
           FHAT(I)=FHAT(I)+P2TP2IP2TP1(I,J)*VEC1(J)
          ENDDO
          FHAT(I)=XMEAN(ITAU-IK+I)-FHAT(I)
C Calculate MSE over validation series over all the k-period-ahead forecasts
          IF (ITRANS.EQ.0) THEN
           MSEMAT(ITAU,IR)=MSEMAT(ITAU,IR)+((F(IN4+I)-FHAT(I))**2)/
     +     DFLOAT(IN2*IK)
          ENDIF
          IF (ITRANS.EQ.1) THEN
           MSEMAT(ITAU,IR)=MSEMAT(ITAU,IR)+((F(IN4+I)**2-
     +     FHAT(I)**2)**2)/DFLOAT(IN2*IK)
          ENDIF
          IF (ITRANS.EQ.2) THEN
           MSEMAT(ITAU,IR)=MSEMAT(ITAU,IR)+((DEXP(F(IN4+I))-
     +     DEXP(FHAT(I)))**2)/DFLOAT(IN2*IK)
          ENDIF
         ENDDO

         IN3=IN3+1
         IN4=IN4+1
  101   CONTINUE
  102  CONTINUE
  103 CONTINUE

C Calculate mean(MSE) and var(MSE)
      DO I=1,IR1
       DO J=1,4
        SORTMSE(I,J)=0.0D0
       ENDDO
      ENDDO

      DO I=1,IR1
C Store number of leading eigenvectors in column [1] of SORTMSE matrix
       SORTMSE(I,1)=I
C Calculate mean(MSE) and store in column [2] of SORTMSE matrix
       DO J=IR1+IK,ITAU1
        SORTMSE(I,2)=SORTMSE(I,2)+MSEMAT(J,I)/DFLOAT(ITAU1-IR1-IK+1)
       ENDDO
C Calculate var(MSE) and store in column [3] of SORTMSE matrix
       DO J=IR1+IK,ITAU1
        SORTMSE(I,3)=SORTMSE(I,3)+((MSEMAT(J,I)-SORTMSE(I,2))**2)/
     +  DFLOAT(ITAU1-IR1-IK)
       ENDDO
C Calculate mean(MSE)^2+var(MSE) and store in column [4] of SORTMSE matrix
       SORTMSE(I,4)=SORTMSE(I,2)**2+SORTMSE(I,3)
      ENDDO

C Sort the results by mean(MSE)^2+var(MSE), i.e. by column [4] of SORTMSE matrix
      DO I=1,IR1
       DO J=I,IR1
        IF (SORTMSE(J,4).LT.SORTMSE(I,4)) THEN
         TEMP1=SORTMSE(I,1)
         TEMP2=SORTMSE(I,2)
         TEMP3=SORTMSE(I,3)
         TEMP4=SORTMSE(I,4)
         SORTMSE(I,1)=SORTMSE(J,1)
         SORTMSE(I,2)=SORTMSE(J,2)
         SORTMSE(I,3)=SORTMSE(J,3)
         SORTMSE(I,4)=SORTMSE(J,4)
         SORTMSE(J,1)=TEMP1
         SORTMSE(J,2)=TEMP2
         SORTMSE(J,3)=TEMP3
         SORTMSE(J,4)=TEMP4
        ENDIF
       ENDDO
      ENDDO

C Select the model with the top model, with lowest mean^2+var
      IMODEL=SORTMSE(1,1)
      MSE1=SORTMSE(1,2)
      MSE2=SORTMSE(1,3)
      MSE3=SORTMSE(1,4)
C Return the minimum window kength (tau) used during FV
      ITAU2=IR1+IK
      END

C *********************************************************************
C                    SIGNALSERIES2 [J. DE KLERK, 05/06/2001]
C *********************************************************************
      SUBROUTINE SIGNALSERIES2(F,INOBS,IN1,ITAU,IR,SIGNAL,XMEAN,PV,
     +ITERATE,ISERIES)
C *********************************************************************
C This subroutine reconstructs signal series
C If ISERIES=0 then no signal reconstuction is performed and the
C original time series is returned
C If ISERIES=1 then a single iteration is performed and the
C approximate series is returned
C If ISERIES=2 the Cadzow-signal series is reconstructed
C ***********INPUT VARIABLES*******************************************
C        F : Vector of Time Series Observations
C    INOBS : Commont length of original multiple series
C      IN1 : Common length of multiple series used for training series
C     ITAU : Number of rows in Trajectory Matrix (tau)
C       IR : Number of eigenvectors in base of Vr
C  ISERIES : If ISERIES=0 No iteration are performed and the signal
C            series returned is the original series with PV,XMEAN
C          : If ISERIES=1 A single iteration is performed and the
C            approximate series with PV and XMEAN is returned
C          : If ISERIES=2 The Cadzow-signal is constructed and PV,
C            XMEAN is returned
C ***********OUTPUT VARIABLES******************************************
C   SIGNAL : smoothed series (reconstructed signal series)
C    XMEAN : shift vector of r-flat L
C       PV : projection matrix of r-flat L
C  ITERATE : number of iterations until convergence
C *********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      DIMENSION F(IMAXOBS),SIGNAL(IMAXOBS)
      DIMENSION X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION XXT(IMAXTAU,IMAXTAU),V(IMAXTAU,IMAXTAU),EVAL(IMAXTAU)
```

```
      DIMENSION PV(IMAXTAU,IMAXTAU),XTILDE(IMAXTAU,IMAXCOLS)
      DIMENSION VEC1(IMAXTAU),VEC2(IMAXTAU),XHAT(IMAXTAU,IMAXCOLS)
      DIMENSION XHAT1(IMAXTAU,IMAXCOLS)

C Step 1 : Store the original time series as the signal series
C Note : Use only the first N1 elements of the series
      DO I=1,INOBS
       SIGNAL(I)=F(I)
      ENDDO
C Construct the signal series iteratively
C Project the column vectors of XTILDE onto linear subspace Vr
C and shift back to r-flat L. Average over reverse diagonals
C Iterate until convergence
      EPSILON=1.0D0
      ITERATE=0
  15  IF (EPSILON.GE.1E-6) THEN
C Set the number of columns in the trajectory matrix
       ICOLS=IN1-ITAU+1
C Create Trajectory Matrix, X
       CALL TRAJECTORYMAT(SIGNAL,ITAU,ICOLS,X)
C Create Centred Trajectory Matrix, XTILDE
       CALL CTRAJECTORYMAT(ITAU,ICOLS,X,XMEAN,XTILDE)
C Create Scatter Matrix, XXT
       CALL SCATMAT(XTILDE,ITAU,ICOLS,XXT)
C Calculate Eigenvectors of Matrix XXT that forms
C orthonormal basis for subspace Vr (IMSL routine)
       CALL DEVCSF(ITAU,XXT,IMAXTAU,EVAL,V,IMAXTAU)
C Determine the rank of matrix XXT, i.e. non-zero eigenvalues
C if rank is less than IR (eigenvectors requested) adjust IR
       IRANK=0
       DO IIR=1,IR
        IF (EVAL(IIR).GT.1E-9) THEN
         IRANK=IRANK+1
        ENDIF
       ENDDO
C Make provision for a series which is a constant
       IF (IRANK.LT.1) THEN
        IRANK=1
       ENDIF
C Create Projection Matrix PV of linear subspace Vr
       CALL PROJECTMAT2(V,ITAU,IRANK,PV)

C Calculate measure to check convergence (distance measure**2)
       SUM1=0.0D0
       DO IK=1,IRANK
        SUM1=SUM1+EVAL(IK)
       ENDDO
       SUM2=0.0D0
       DO IK=IRANK+1,ITAU
        SUM2=SUM2+EVAL(IK)
       ENDDO
       EPSILON=SUM2/SUM1

C The following statement is used to sidestep the signal reconstructive part if
C if the original series must be used
       IF (ISERIES.EQ.0) THEN
        EPSILON=1E-8
       ENDIF
C The following statement is used to perform a single Cadzow iteration
       IF ((ISERIES.EQ.1).AND.(ITERATE.EQ.1)) THEN
        EPSILON=1E-8
       ENDIF

       IF (EPSILON.GE.1E-6) THEN
```

```
        ITERATE=ITERATE+1
C Create a new matrix XHAT by projecting the column vectors of
C matrix XTILDE onto Vr and shifting to r-flat L
        DO J=1,ICOLS
         DO I=1,ITAU
          SUM=0.0D0
          DO IJ=1,ITAU
           SUM=SUM+PV(I,IJ)*XTILDE(IJ,J)
          ENDDO
          XHAT(I,J)=SUM+XMEAN(I)
         ENDDO
        ENDDO

C Average over the reverse diagonals of matrix XHAT
C See algorithms in Danilov p.12
C Reconstruct f(1),...,f(tau)
        DO I=1,ITAU
         SIGNAL(I)=0.0D0
         DO J=1,I
          SIGNAL(I)=SIGNAL(I)+XHAT(J,I-J+1)/I
         ENDDO
        ENDDO
C Reconstruct f(tau+1),...,f(n)
        DO I=ITAU+1,ICOLS
         SIGNAL(I)=0.0D0
         DO J=1,ITAU
          SIGNAL(I)=SIGNAL(I)+XHAT(J,I-J+1)/ITAU
         ENDDO
        ENDDO
C Reconstruct f(n+1),...,f(N)
        DO I=ICOLS+1,IN1
         SIGNAL(I)=0.0D0
         DO J=1,(IN1-I+1)
          SIGNAL(I)=SIGNAL(I)+XHAT(J+I-ICOLS,ICOLS-J+1)/(IN1-I+1)
         ENDDO
        ENDDO
       ENDIF
C Exit the loop if more than 1000 iterations experienced
       IF (ITERATE.GT.1000) THEN
           EPSILON=1E-8
       ENDIF
       GO TO 15
      ENDIF
      RETURN
      END

C ********************************************************************
C                  FORECAST1 [J. DE KLERK, 27/01/2002]
C ********************************************************************
      SUBROUTINE FORECAST1(F,INOBS,ISTART,ITAU,IR,IEVEC,IK,ICENT,ICH1,
     +FHAT,MAD,SSE,MSE,MAPE,ITRANS)
C ********************************************************************
C This subroutine forecasts a time series using SSA
C The forecasts are 1-period-ahead forecasts without re-estimtion
C If in-sample forecasts are genereated the forecast and not the
C original in-sample time series are used to generate next forecasts
C ***********INPUT VARIABLES******************************************
C        F : Vector containing original time series
C           IN : Length of original time series
C ***********OUTPUT VARIABLES*****************************************
C      IN1 : Length of training time series
C      IN2 : Length of validation time series
C  IMAXTAU : Maximum window length looped over
C    IMAXR : Maximum number of leading eigenvectors looped over
```

```
C       IK : Period-ahead forecasting used during FV
C       IR : Model selected using Forward Validation (FV)
C            Returns the number of eigenvectors
C        X : Trajectory Matrix
C   XTILDE : Centred Trajectory Matrix
C      XXT : Scatter matrix of matrix XTILDE
C        V : Matrix of eigenvectors of the scatter matrix XXT
C     EVAL : Vector of eigenvalues of the scatter matrix XXT
C       PV : Projection matrix of parallel linear subspace Vr
C        P : Projection matrix of orthogonal parallel linear subspace
C       P1 : Partitioning of matrix P
C       P2 : Partitioning of matrix P
C    P2TP2 : Matrix P2'*P2
C    P2TP1 : Matrix P2'*P1
C P2TP2INV : Matrix inv(P2'P2)
C    XMEAN : Shift vector to r-flat
C ********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      DIMENSION F(IMAXOBS),X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION SIGNAL(IMAXOBS)
      DIMENSION XTILDE(IMAXTAU,IMAXCOLS),XXT(IMAXTAU,IMAXTAU)
      DIMENSION PV(IMAXTAU,IMAXTAU)
      DIMENSION P1(IMAXTAU,IMAXTAU),P2(IMAXTAU,IMAXTAU)
      DIMENSION P(IMAXTAU,IMAXTAU),P2TP2(200,200)
      DIMENSION P2TP1(IMAXTAU,IMAXTAU),P2TP2INV(200,200)
      DIMENSION P2TP2IP2TP1(IMAXTAU,IMAXTAU)
      DIMENSION VEC1(IMAXTAU),FHAT(IMAXOBS)
      DIMENSION EVAL(IMAXTAU),V(IMAXTAU,IMAXTAU)

C Open file 'C:\SHONG.DAT' which contains % variation explained
      OPEN(UNIT=4,FILE='C:\SHONG.DAT')
C Rest measure of out-of-sample forecasting accuracy
      MAD=0.0D0
      SSE=0.0D0
      MSE=0.0D0
      MAPE=0.0D0

C Determine number of in-sample and out-of-sample forecasts
C IT1 is the index position of series to use during unfolding
C IFORE1 is the number of in-sample-forecasts
      IF (ISTART.LT.INOBS) THEN
       IT1=(ISTART-1)
       IFORE1=INOBS-ISTART+1
C Check that supplied number of columns can be used
C Re-set number of columns in trajectory matrix
C Note that the number of columns statement below is not placed
C in the loop structure. It would not remain constant in that case.
       IF (ITAU.GT.INT((IT1+1)/2)) THEN
        ITAU=INT((IT1+1)/2)
       ENDIF
       ICOLS=IT1-ITAU+1
      ELSE
C Remember to check number of columns here as well
       IT1=INOBS
       IF(ITAU.GT.INT((INOBS+1)/2)) THEN
        ITAU=INT((INOBS+1)/2)
       ENDIF
       ICOLS=INOBS-ITAU+1
      ENDIF

C [Use reconstructed Cadzow signal series]
       IF (ICH1.EQ.3) THEN
        CALL CADZOWSIGNAL(F,IT1,ITAU,IR,IEVEC,ICENT,SIGNAL,XMEAN,
     +  PV,ITERATE)
C The Cadzow routine returns the signal in vector SIGNAL and also
C returns the shift vector and projection matrix
       ENDIF

C [Use approximate series]
       IF (ICH1.EQ.2) THEN
        CALL APPROXSIGNAL(F,IT1,ITAU,IR,IEVEC,ICENT,SIGNAL,XMEAN,PV)
C The routine returns the signal in vector SIGNAL and also
C returns the shift vector and projection matrix
       ENDIF

C [If original series must be used]
       IF (ICH1.EQ.1) THEN
        DO I=1,INOBS
         SIGNAL(I)=F(I)
        ENDDO
C Create Trajectory Matrix, X
        CALL TMAT(SIGNAL,ITAU,ICOLS,X)
C Create Centred Trajectory Matrix, XTILDE
        CALL CTMAT(ITAU,ICOLS,X,XMEAN,ICENT,XTILDE)
C Create Scatter Matrix, XXT
        CALL SCATMAT(XTILDE,ITAU,ICOLS,XXT)
C Calculate Eigenvectors of Matrix XXT that forms
C orthonormal basis for subspace Vr (IMSL routine)
        CALL DEVCSF(ITAU,XXT,IMAXTAU,EVAL,V,IMAXTAU)
C Calculate % variation explained by eigenvectors
        SUM1=0.0D0
        DO II=1,ITAU
         SUM1=SUM1+EVAL(II)
        ENDDO
        DO II=1,ITAU
         WRITE(4,'(G30.15)')EVAL(II)/SUM1
        ENDDO
        CLOSE(4)
C Calculate percentage variation explained
C Determine the rank of matrix XXT, i.e. non-zero eigenvalues
C if rank is less than IR (eigenvectors requested) adjust IR
        IRANK=0
        DO IIR=1,IR
         IF (EVAL(IIR).GT.1E-9) THEN
          IRANK=IRANK+1
         ENDIF
        ENDDO
        ICONST=0
C Make provision for a series which is a constant
        IF (IRANK.LT.1) THEN
        IRANK=1
         ICONST=1
        ENDIF
        IF (IRANK.GT.IR) THEN
         IRANK=IR
        ENDIF
C Create Projection Matrix PV of linear subspace Vr
        CALL PROJECTMAT(V,ITAU,IRANK,IEVEC,PV)
       ENDIF

C For all combinations ensure that the rest of original series
C is in SIGNAL vector
       IF (ISTART.LT.INOBS) THEN
        DO I=IT1+1,INOBS
         SIGNAL(I)=F(I)
        ENDDO
       ENDIF

C Now that the correct SIGNAL, projection matrix PV is available
C Create projection matrix of orthogonal parallel linear subspace Vr
       DO I=1,ITAU
        DO J=I,ITAU
         IF (I.EQ.J) THEN
          P(I,J)=1.0D0-PV(I,J)
         ELSE
          P(I,J)=0.0D0-PV(I,J)
         ENDIF
         P(J,I)=P(I,J)
        ENDDO
       ENDDO

C Partition projection matrix P into P1 and P2
       DO I=1,ITAU
        DO J=1,ITAU
         IF (J.LE.(ITAU-1)) THEN
          P1(I,J)=P(I,J)
         ENDIF
         IF (J.GE.(ITAU-1)+1) THEN
          P2(I,J-(ITAU-1))=P(I,J)
         ENDIF
        ENDDO
       ENDDO

C Create P2'P2 and invert the result
       DO I=1,1
        DO J=1,1
         P2TP2(I,J)=0.0D0
         DO IJ=1,ITAU
          P2TP2(I,J)=P2TP2(I,J)+P2(IJ,I)*P2(IJ,J)
         ENDDO
         P2TP2(J,I)=P2TP2(I,J)
        ENDDO
       ENDDO

C Set up inverse of P2TP2
       CALL DLINDS(1,P2TP2,200,P2TP2INV,200)

C Create P2'P1 and invert the result
       DO I=1,1
        DO J=1,(ITAU-1)
         P2TP1(I,J)=0.0D0
         DO IJ=1,ITAU
          P2TP1(I,J)=P2TP1(I,J)+P2(IJ,I)*P1(IJ,J)
         ENDDO
        ENDDO
       ENDDO

C Create inv(P2'P2)P2TP1
       DO I=1,1
        DO J=1,(ITAU-1)
         P2TP2IP2TP1(I,J)=0.0D0
         DO IJ=1,1
          P2TP2IP2TP1(I,J)=P2TP2IP2TP1(I,J)+P2TP2INV(I,IJ)*P2TP1(IJ,J)
         ENDDO
        ENDDO
       ENDDO

C The loop that generates IK recurrent 1-period-ahead forecsats starts
      DO 101 I1=1,IK
C Create k-step-ahead forecasts for the multiple time series
C The vector SIGNAL is used and will contain either the 1)original series
C 2) approximate series or 3)Cadzow signal series
       DO II=1,ITAU-1
        IF (ICENT.EQ.1) THEN
         VEC1(II)=SIGNAL(IT1-ITAU+1+II)-XMEAN(II)
        ELSE
         VEC1(II)=SIGNAL(IT1-ITAU+1+II)
        ENDIF
       ENDDO

       FHAT(I1)=0.0D0
       DO J=1,(ITAU-1)
        FHAT(I1)=FHAT(I1)+P2TP2IP2TP1(1,J)*VEC1(J)
       ENDDO
       IF (ICENT.EQ.1) THEN
        FHAT(I1)=XMEAN((ITAU-1)+1)-FHAT(I1)
       ELSE
        FHAT(I1)=-FHAT(I1)
       ENDIF

C Here we concatenate the one-period-ahead forecast to our oiginal series
C Hence we consider it as if it is the "new' reconstructed value and use it
C This is in line with the true spirit of recurrent forecasts
       SIGNAL(IT1+1)=FHAT(I1)

       IF (IT1+1.LE.INOBS) THEN
C Calculate measures of out-of-sample forecasting accuracy
        IF (ITRANS.EQ.1) THEN
         MAD=MAD+DABS(F(IT1+1)-FHAT(I1))/
     +   DFLOAT(IFORE1)
         SSE=SSE+((F(IT1+1)-FHAT(I1))**2)
         MSE=MSE+((F(IT1+1)-FHAT(I1))**2)/
     +   DFLOAT(IFORE1)
         MAPE=MAPE+DABS(F(IT1+1)-FHAT(I1))/
     +    (DFLOAT(IFORE1)*F(IT1+1))*100
        ENDIF
        IF (ITRANS.EQ.2) THEN
         MAD=MAD+DABS(EXP(F(IT1+1))-EXP(FHAT(I1)))/
     +   DFLOAT(IFORE1)
         SSE=SSE+((EXP(F(IT1+1))-EXP(FHAT(I1)))**2)
         MSE=MSE+((EXP(F(IT1+1))-EXP(FHAT(I1)))**2)/
     +   DFLOAT(IFORE1)
         MAPE=MAPE+DABS(EXP(F(IT1+1))-EXP(FHAT(I1)))/
     +    (DFLOAT(IFORE1)*EXP(F(IT1+1)))*100
        ENDIF
        IF (ITRANS.EQ.3) THEN
         MAD=MAD+DABS((F(IT1+1)**2)-(FHAT(I1)**2))/
     +   DFLOAT(IFORE1)
         SSE=SSE+(((F(IT1+1)**2)-(FHAT(I1)**2))**2)
         MSE=MSE+(((F(IT1+1)**2)-(FHAT(I1)**2))**2)/
     +   DFLOAT(IFORE1)
         MAPE=MAPE+DABS((F(IT1+1)**2)-(FHAT(I1)**2))/
     +    (DFLOAT(IFORE1)*(F(IT1+1)**2))*100
        ENDIF
       ENDIF
C Increase training series length by one
       IT1=IT1+1
  101  CONTINUE
      END
```

```
C ********************************************************************
C                   FORECAST2 [J. DE KLERK, 2002]
C ********************************************************************
      SUBROUTINE FORECAST2(F,INOBS,ISTART,ITAU,IR,IEVEC,IK,ICENT,ICH1,
     +FHAT,MAD,SSE,MSE,MAPE,ITRANS)
C ********************************************************************
C This subroutine forecasts a time series using SSA
C The forecasts are 1-period-ahead forecasts with re-estimtion
C If in-sample forecasts are genereated the forecast and not the
C original in-sample time series are used to generate next forecasts
C ***********INPUT VARIABLES******************************************
C        F : Vector containing original time series
C       IN : Length of original time series
C ***********OUTPUT VARIABLES*****************************************
C      IN1 : Length of training time series
C      IN2 : Length of validation time series
C  IMAXTAU : Maximum window length looped over
C    IMAXR : Maximum number of leading eigenvectors looped over
C       IK : Period-ahead forecasting used during FV
C       IR : Model selected using Forward Validation (FV)
C            Returns the number of eigenvectors
C        X : Trajectory Matrix
C   XTILDE : Centred Trajectory Matrix
C      XXT : Scatter matrix of matrix XTILDE
C        V : Matrix of eigenvectors of the scatter matrix XXT
C     EVAL : Vector of eigenvalues of the scatter matrix XXT
C       PV : Projection matrix of parallel linear subspace Vr
C        P : Projection matrix of orthogonal parallel linear subspace
C       P1 : Partitioning of matrix P
C       P2 : Partitioning of matrix P
C    P2TP2 : Matrix P2'*P2
C    P2TP1 : Matrix P2'*P1
C P2TP2INV : Matrix inv(P2'P2)
C    XMEAN : Shift vector to r-flat
C ********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      DIMENSION F(IMAXOBS),X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION SIGNAL(IMAXOBS)
      DIMENSION XTILDE(IMAXTAU,IMAXCOLS),XXT(IMAXTAU,IMAXTAU)
      DIMENSION PV(IMAXTAU,IMAXTAU)
      DIMENSION P1(IMAXTAU,IMAXTAU),P2(IMAXTAU,IMAXTAU)
      DIMENSION P(IMAXTAU,IMAXTAU),P2TP2(200,200)
      DIMENSION P2TP1(IMAXTAU,IMAXTAU),P2TP2INV(200,200)
      DIMENSION P2TP2IP2TP1(IMAXTAU,IMAXTAU)
      DIMENSION VEC1(IMAXTAU),FHAT(IMAXOBS)
      DIMENSION EVAL(IMAXTAU),V(IMAXTAU,IMAXTAU)

C Rest measure of out-of-sample forecasting accuracy
      MAD=0.0D0
      SSE=0.0D0
      MSE=0.0D0
      MAPE=0.0D0

C Determine number of in-sample and out-of-sample forecasts
C IT1 is the index position of series to use during unfolding
C IFORE1 is the number of in-sample-forecasts
```

```
      IF (ISTART.LT.INOBS) THEN
       IT1=(ISTART-1)
       IFORE1=INOBS-ISTART+1
C Check that supplied number of columns can be used
C Re-set number of columns in trajectory matrix
C Note that the number of columns statement below is not placed
C in the loop structure. It would not remain constant in that case.
       IF (ITAU.GT.INT((IT1+1)/2)) THEN
        ITAU=INT((IT1+1)/2)
       ENDIF
       ICOLS=IT1-ITAU+1
      ELSE
C Remember to check number of columns here as well
       IT1=INOBS
       IF(ITAU.GT.INT((INOBS+1)/2)) THEN
        ITAU=INT((INOBS+1)/2)
       ENDIF
       ICOLS=INOBS-ITAU+1
      ENDIF


C The loop that generates IK recurrent 1-period-ahead forecsats starts
      DO 101 I1=1,IK
C [Use reconstructed Cadzow signal series]
       IF (ICH1.EQ.3) THEN
        CALL CADZOWSIGNAL(F,IT1,ITAU,IR,IEVEC,ICENT,SIGNAL,XMEAN,
     +  PV,ITERATE)
C The Cadzow routine returns the signal in vector SIGNAL and also
C returns the shift vector and projection matrix
       ENDIF

C [Use approximate series]
       IF (ICH1.EQ.2) THEN
        CALL APPROXSIGNAL(F,IT1,ITAU,IR,IEVEC,ICENT,SIGNAL,XMEAN,PV)
C The routine returns the signal in vector SIGNAL and also
C returns the shift vector and projection matrix
       ENDIF

C [If original series must be used]
       IF (ICH1.EQ.1) THEN
        DO I=1,INOBS
         SIGNAL(I)=F(I)
        ENDDO
C Create Trajectory Matrix, X
        CALL TMAT(SIGNAL,ITAU,ICOLS,X)
C Create Centred Trajectory Matrix, XTILDE
        CALL CTMAT(ITAU,ICOLS,X,XMEAN,ICENT,XTILDE)
C Create Scatter Matrix, XXT
        CALL SCATMAT(XTILDE,ITAU,ICOLS,XXT)
C Calculate Eigenvectors of Matrix XXT that forms
C orthonormal basis for subspace Vr (IMSL routine)
        CALL DEVCSF(ITAU,XXT,IMAXTAU,EVAL,V,IMAXTAU)
C Determine the rank of matrix XXT, i.e. non-zero eigenvalues
C if rank is less than IR (eigenvectors requested) adjust IR
        IRANK=0
        DO IIR=1,IR
         IF (EVAL(IIR).GT.1E-9) THEN
          IRANK=IRANK+1
         ENDIF
        ENDDO
        ICONST=0
C Make provision for a series which is a constant
        IF (IRANK.LT.1) THEN
        IRANK=1
         ICONST=1
        ENDIF
        IF (IRANK.GT.IR) THEN
         IRANK=IR
        ENDIF
C Create Projection Matrix PV of linear subspace Vr
        CALL PROJECTMAT(V,ITAU,IRANK,IEVEC,PV)
       ENDIF

C For all combinations ensure that the rest of original series
C is in SIGNAL vector
       IF (ISTART.LT.INOBS) THEN
        DO I=IT1+1,INOBS
         SIGNAL(I)=F(I)
        ENDDO
       ENDIF

C Now that the correct SIGNAL, projection matrix PV is available
C Create projection matrix of orthogonal parallel linear subspace Vr
       DO I=1,ITAU
        DO J=I,ITAU
         IF (I.EQ.J) THEN
          P(I,J)=1.0D0-PV(I,J)
         ELSE
          P(I,J)=0.0D0-PV(I,J)
         ENDIF
         P(J,I)=P(I,J)
        ENDDO
       ENDDO

C Partition projection matrix P into P1 and P2
       DO I=1,ITAU
        DO J=1,ITAU
         IF (J.LE.(ITAU-1)) THEN
          P1(I,J)=P(I,J)
         ENDIF
         IF (J.GE.(ITAU-1)+1) THEN
          P2(I,J-(ITAU-1))=P(I,J)
         ENDIF
        ENDDO
       ENDDO

C Create P2'P2 and invert the result
       DO I=1,1
        DO J=1,1
         P2TP2(I,J)=0.0D0
         DO IJ=1,ITAU
          P2TP2(I,J)=P2TP2(I,J)+P2(IJ,I)*P2(IJ,J)
         ENDDO
         P2TP2(J,I)=P2TP2(I,J)
        ENDDO
       ENDDO

C Set up inverse of P2TP2
       CALL DLINDS(1,P2TP2,200,P2TP2INV,200)

C Create P2'P1 and invert the result
       DO I=1,1
        DO J=1,(ITAU-1)
         P2TP1(I,J)=0.0D0
         DO IJ=1,ITAU
          P2TP1(I,J)=P2TP1(I,J)+P2(IJ,I)*P1(IJ,J)
         ENDDO
        ENDDO
       ENDDO

C Create inv(P2'P2)P2TP1
       DO I=1,1
        DO J=1,(ITAU-1)
         P2TP2IP2TP1(I,J)=0.0D0
         DO IJ=1,1
          P2TP2IP2TP1(I,J)=P2TP2IP2TP1(I,J)+P2TP2INV(I,IJ)*P2TP1(IJ,J)
         ENDDO
        ENDDO
       ENDDO

C Create k-step-ahead forecasts for the multiple time series
C The vector SIGNAL is used and will contain either the 1)original series
C 2) approximate series or 3)Cadzow signal series
       DO II=1,ITAU-1
        IF (ICENT.EQ.1) THEN
         VEC1(II)=SIGNAL(IT1-ITAU+1+II)-XMEAN(II)
        ELSE
         VEC1(II)=SIGNAL(IT1-ITAU+1+II)
        ENDIF
       ENDDO

       FHAT(I1)=0.0D0
       DO J=1,(ITAU-1)
        FHAT(I1)=FHAT(I1)+P2TP2IP2TP1(1,J)*VEC1(J)
       ENDDO
       IF (ICENT.EQ.1) THEN
        FHAT(I1)=XMEAN((ITAU-1)+1)-FHAT(I1)
       ELSE
        FHAT(I1)=-FHAT(I1)
       ENDIF

C Here we concatenate the one-period-ahead forecast to our oiginal series
C Hence we consider it as if it is the "new' reconstructed value and use it
C This is in line with the true spirit of recurrent forecasts
       SIGNAL(IT1+1)=FHAT(I1)

C Calculate measures of out-of-sample forecasting accuracy
       IF (IT1+1.LE.INOBS) THEN
        IF (ITRANS.EQ.1) THEN
         MAD=MAD+DABS(F(IT1+1)-FHAT(I1))/
     +   DFLOAT(IFORE1)
         SSE=SSE+((F(IT1+1)-FHAT(I1))**2)
         MSE=MSE+((F(IT1+1)-FHAT(I1))**2)/
     +   DFLOAT(IFORE1)
         MAPE=MAPE+DABS(F(IT1+1)-FHAT(I1))/
     +   (DFLOAT(IFORE1)*F(IT1+1))*100
        ENDIF
        IF (ITRANS.EQ.2) THEN
         MAD=MAD+DABS(DEXP(F(IT1+1))-DEXP(FHAT(I1)))/
     +   DFLOAT(IFORE1)
         SSE=SSE+((DEXP(F(IT1+1))-DEXP(FHAT(I1)))**2)
         MSE=MSE+((DEXP(F(IT1+1))-DEXP(FHAT(I1)))**2)/
     +   DFLOAT(IFORE1)
         MAPE=MAPE+DABS(DEXP(F(IT1+1))-DEXP(FHAT(I1)))/
     +   (DFLOAT(IFORE1)*DEXP(F(IT1+1)))*100
        ENDIF
        IF (ITRANS.EQ.3) THEN
         MAD=MAD+DABS((F(IT1+1)**2)-(FHAT(I1)**2))/
     +   DFLOAT(IFORE1)
         SSE=SSE+(((F(IT1+1)**2)-(FHAT(I1)**2))**2)
         MSE=MSE+(((F(IT1+1)**2)-(FHAT(I1)**2))**2)/
     +   DFLOAT(IFORE1)
```

```
         MAPE=MAPE+DABS((F(IT1+1)**2)-(FHAT(I1)**2))/
     +   (DFLOAT(IFORE1)*(F(IT1+1)**2))*100
        ENDIF
       ENDIF
C Increase training series length by one
       IT1=IT1+1
  101  CONTINUE
      END


C *******************************************************************
C                  FORECAST3 [J. DE KLERK, 2002]
C *******************************************************************
      SUBROUTINE FORECAST3(F,INOBS,ISTART,ITAU,IR,IEVEC,IK,ICENT,ICH1,
     +FHAT,MAD,SSE,MSE,MAPE,ITRANS)
C *******************************************************************
C This subroutine generates joint-horizon forecasts
C ***********INPUT VARIABLES*****************************************
C        F : Vector containing original time series
C    INOBS : Length of original time series
C ***********OUTPUT VARIABLES****************************************
C  IMAXTAU : Maximum window length looped over
C    IMAXR : Maximum number of leading eigenvectors looped over
C   IFORE1 : Number of in-sample forecasts generated
C   IFORE2 : Number of out-of-sample forecasts generated
C      IT1 : Length of series to use when creating signal series
C       IK : Period-ahead forecasting used during FV
C       IR : Model selected using Forward Validation (FV)
C            Returns the number of eigenvectors
C        X : Trajectory Matrix
C   XTILDE : Centred Trajectory Matrix
C      XXT : Scatter matrix of matrix XTILDE
C        V : Matrix of eigenvectors of the scatter matrix XXT
C     EVAL : Vector of eigenvalues of the scatter matrix XXT
C       PV : Projection matrix of parallel linear subspace Vr
C        P : Projection matrix of orthogonal parallel linear subspace
C       P1 : Partitioning of matrix P
C       P2 : Partitioning of matrix P
C    P2TP2 : Matrix P2'*P2
C    P2TP1 : Matrix P2'*P1
C P2TP2INV : Matrix inv(P2'P2)
C    XMEAN : Shift vector to r-flat
C *******************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      DIMENSION F(IMAXOBS),X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION SIGNAL(IMAXOBS),FHAT(200)
      DIMENSION XTILDE(IMAXTAU,IMAXCOLS),XXT(IMAXTAU,IMAXTAU)
      DIMENSION PV(IMAXTAU,IMAXTAU)
      DIMENSION P1(IMAXTAU,IMAXTAU),P2(IMAXTAU,IMAXTAU)
      DIMENSION P(IMAXTAU,IMAXTAU),P2TP2(200,200)
      DIMENSION P2TP1(IMAXTAU,IMAXTAU),P2TP2INV(200,200)
      DIMENSION P2TP2IP2TP1(IMAXTAU,IMAXTAU)
      DIMENSION VEC1(IMAXTAU)
      DIMENSION EVAL(IMAXTAU),V(IMAXTAU,IMAXTAU)

C Open file 'C:\SHONG.DAT' which contains % variation explained
      OPEN(UNIT=4,FILE='C:\SHONG.DAT')
```

```
C Determine number of in-sample and out-of-sample forecasts
C Also set index position of series to use
      IF (ISTART.LT.INOBS) THEN
       IT1=(ISTART-1)
       IFORE1=INOBS-ISTART+1
       IF ((ISTART+IK).GT.INOBS) THEN
        IFORE2=(ISTART+IK)-INOBS
       ENDIF
       ELSE
       IT1=INOBS
      ENDIF
C Set number of columns in trajectory matrix
C IT1 is length of time series to use to forecast with
      ICOLS=IT1-ITAU+1

C [Use reconstructed Cadzow signal series]
      IF (ICH1.EQ.3) THEN
       CALL CADZOWSIGNAL(F,IT1,ITAU,IR,IEVEC,ICENT,SIGNAL,XMEAN,
     + PV,ITERATE)
C The Cadzow routine returns the signal in vector SIGNAL and also
C returns the shift vector and projection matrix
      ENDIF

C [Use approximate series]
      IF (ICH1.EQ.2) THEN
       CALL APPROXSIGNAL(F,IT1,ITAU,IR,IEVEC,ICENT,SIGNAL,XMEAN,PV)
C The routine returns the signal in vector SIGNAL and also
C returns the shift vector and projection matrix
      ENDIF

C [If original series must be used]
      IF (ICH1.EQ.1) THEN
       DO I=1,INOBS
        SIGNAL(I)=F(I)
       ENDDO
C Create Trajectory Matrix, X
       CALL TMAT(F,ITAU,ICOLS,X)
C Create Centred Trajectory Matrix, XTILDE
       CALL CTMAT(ITAU,ICOLS,X,XMEAN,ICENT,XTILDE)
C Create Scatter Matrix, XXT
       CALL SCATMAT(XTILDE,ITAU,ICOLS,XXT)
C Calculate Eigenvectors of Matrix XXT that forms
C orthonormal basis for subspace Vr (IMSL routine)
       CALL DEVCSF(ITAU,XXT,IMAXTAU,EVAL,V,IMAXTAU)
C Calculate % variation explained by eigenvectors
        SUM1=0.0D0
        DO II=1,ITAU
         SUM1=SUM1+EVAL(II)
        ENDDO
        DO II=1,ITAU
         WRITE(4,'(G30.15)')EVAL(II)/SUM1
        ENDDO
        CLOSE(4)
C Determine the rank of matrix XXT, i.e. non-zero eigenvalues
C if rank is less than IR (eigenvectors requested) adjust IR
       IRANK=0
       DO IIR=1,IR
        IF (EVAL(IIR).GT.1E-9) THEN
         IRANK=IRANK+1
        ENDIF
       ENDDO
       ICONST=0
C Make provision for a series which is a constant
```

```
       IF (IRANK.LT.1) THEN
       IRANK=1
        ICONST=1
       ENDIF
       IF (IRANK.GT.IR) THEN
        IRANK=IR
       ENDIF
C Create Projection Matrix PV of linear subspace Vr
       CALL PROJECTMAT(V,ITAU,IRANK,IEVEC,PV)
      ENDIF


C Now that the correct SIGNAL, projection matrix PV is available
C Create projection matrix of orthogonal parallel linear subspace Vr
      DO I=1,ITAU
       DO J=I,ITAU
        IF (I.EQ.J) THEN
         P(I,J)=1.0D0-PV(I,J)
        ELSE
         P(I,J)=0.0D0-PV(I,J)
        ENDIF
        P(J,I)=P(I,J)
       ENDDO
      ENDDO

C Partition projection matrix P into P1 and P2
      DO I=1,ITAU
       DO J=1,ITAU
        IF (J.LE.(ITAU-IK)) THEN
         P1(I,J)=P(I,J)
        ENDIF
        IF (J.GE.(ITAU-IK)+1) THEN
         P2(I,J-(ITAU-IK))=P(I,J)
        ENDIF
       ENDDO
      ENDDO

C Create P2'P2 and invert the result
      DO I=1,IK
       DO J=1,IK
        P2TP2(I,J)=0.0D0
        DO IJ=1,ITAU
         P2TP2(I,J)=P2TP2(I,J)+P2(IJ,I)*P2(IJ,J)
        ENDDO
        P2TP2(J,I)=P2TP2(I,J)
       ENDDO
      ENDDO

C Set up inverse of P2TP2
      CALL DLINDS(IK,P2TP2,200,P2TP2INV,200)

C Create P2'P1 and invert the result
      DO I=1,IK
       DO J=1,(ITAU-IK)
        P2TP1(I,J)=0.0D0
        DO IJ=1,ITAU
         P2TP1(I,J)=P2TP1(I,J)+P2(IJ,I)*P1(IJ,J)
        ENDDO
       ENDDO
      ENDDO

C Create inv(P2'P2)P2TP1
      DO I=1,IK
       DO J=1,(ITAU-IK)
```

```
        P2TP2IP2TP1(I,J)=0.0D0
        DO IJ=1,IK
         P2TP2IP2TP1(I,J)=P2TP2IP2TP1(I,J)+P2TP2INV(I,IJ)*P2TP1(IJ,J)
        ENDDO
       ENDDO
      ENDDO

C Create k-step-ahead forecasts for the multiple time series
C The vector SIGNAL is used and will contain either the 1)original series
C 2) approximate series or 3)Cadzow signal series
      DO II=1,ITAU-IK
       IF (ICENT.EQ.1) THEN
        VEC1(II)=SIGNAL(IT1-(ITAU-IK)+II)-XMEAN(II)
       ELSE
        VEC1(II)=SIGNAL(IT1-(ITAU-IK)+II)
       ENDIF
      ENDDO

      DO I=1,IK
       FHAT(I)=0.0D0
       DO J=1,(ITAU-IK)
        FHAT(I)=FHAT(I)+P2TP2IP2TP1(I,J)*VEC1(J)
       ENDDO
       IF (ICENT.EQ.1) THEN
        FHAT(I)=XMEAN((ITAU-IK)+I)-FHAT(I)
       ELSE
        FHAT(I)=-1.0D0*FHAT(I)
       ENDIF
      ENDDO

C Calculate measures of out-of-sample forecasting accuracy
C Rest measure of out-of-sample forecasting accuracy
      MAD=0.0D0
      SSE=0.0D0
      MSE=0.0D0
      MAPE=0.0D0
      IF (ISTART.LT.INOBS) THEN
       IF (ITRANS.EQ.1) THEN
        DO I=1,IFORE1
         MAD=MAD+DABS(F(IT1+I)-FHAT(I))/
     +   DFLOAT(IFORE1)
         SSE=SSE+((F(IT1+I)-FHAT(I))**2)
         MSE=MSE+((F(IT1+I)-FHAT(I))**2)/
     +   DFLOAT(IFORE1)
         MAPE=MAPE+DABS(F(IT1+I)-FHAT(I))/
     +   (DFLOAT(IFORE1)*F(IT1+I))*100
        ENDDO
       ENDIF
       IF (ITRANS.EQ.2) THEN
        DO I=1,IFORE1
         MAD=MAD+DABS(DEXP(F(IT1+I))-DEXP(FHAT(I)))/
     +   DFLOAT(IFORE1)
         SSE=SSE+((DEXP(F(IT1+I))-DEXP(FHAT(I)))**2)
         MSE=MSE+((DEXP(F(IT1+I))-DEXP(FHAT(I)))**2)/
     +   DFLOAT(IFORE1)
         MAPE=MAPE+DABS(DEXP(F(IT1+I))-DEXP(FHAT(I)))/
     +   (DFLOAT(IFORE1)*DEXP(F(IT1+I)))*100
        ENDDO
       ENDIF
       IF (ITRANS.EQ.3) THEN
        DO I=1,IFORE1
         MAD=MAD+DABS((F(IT1+I)**2)-(FHAT(I)**2))/
     +   DFLOAT(IFORE1)
         SSE=SSE+(((F(IT1+I)**2)-(FHAT(I)**2))**2)
         MSE=MSE+(((F(IT1+I)**2)-(FHAT(I)**2))**2)/
     +   DFLOAT(IFORE1)
         MAPE=MAPE+DABS((F(IT1+I)**2)-(FHAT(I)**2))/
     +   (DFLOAT(IFORE1)*(F(IT1+I)**2))*100
        ENDDO
       ENDIF
      ENDIF
      END


C ********************************************************************
C                  TMAT [J. DE KLERK, 1999]
C ********************************************************************
      SUBROUTINE TMAT(F,ITAU,ICOLS,X)
C ********************************************************************
C This subroutine creates the Trajectory Matrix [X]
C ***********INPUT VARIABLES******************************************
C        F : Vector of Time Series Observations
C     ITAU : Number of rows in Trajectory Matrix (tau)
C    ICOLS : Number of columns Trajectory Matrix (n=N-tau+1)
C ***********OUTPUT VARIABLES*****************************************
C        X : Trajectory Matrix
C ********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)
      DIMENSION F(IMAXOBS),X(IMAXTAU,IMAXCOLS)

      DO I=1,ITAU
       DO J=1,ICOLS
        X(I,J)=F(I+J-1)
       ENDDO
      ENDDO
      RETURN
      END

C ********************************************************************
C                 CTMAT [J. DE KLERK, 1999]
C ********************************************************************
      SUBROUTINE CTMAT(ITAU,ICOLS,X,XMEAN,ICENT,XTILDE)
C ********************************************************************
C This subroutine creates the Centred Trajectory Matrix [XTILDE]
C ***********INPUT VARIABLES******************************************
C     ITAU : Number of rows in Trajectory Matrix (tau)
C    ICOLS : Number of columns Trajectory Matrix (n=N-tau+1)
C        X : Trajectory Matrix
C    ICENT : If ICENT=1 centring is performed else not
C ***********OUTPUT VARIABLES*****************************************
C    XMEAN : Vector of matrix X row means (shift vector)
C   XTILDE : Centred Trajectory Matrix
C ********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)
      DIMENSION X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION XTILDE(IMAXTAU,IMAXCOLS)
```

```
C If centring is required calculate the mean vector as shift vector
      IF (ICENT.EQ.1) THEN
       DO I=1,ITAU
        XMEAN(I)=0.0D0
        DO J=1,ICOLS
         XMEAN(I)=XMEAN(I)+X(I,J)/DFLOAT(ICOLS)
        ENDDO
        DO J=1,ICOLS
         XTILDE(I,J)=X(I,J)-XMEAN(I)
        ENDDO
       ENDDO
      ELSE
       DO I=1,ITAU
        DO J=1,ICOLS
         XTILDE(I,J)=X(I,J)
        ENDDO
       ENDDO
      ENDIF
      RETURN
      END


C *********************************************************************
C                    SCATMAT [J. DE KLERK, 1999]
C *********************************************************************
      SUBROUTINE SCATMAT(XTILDE,ITAU,ICOLS,XXT)
C *********************************************************************
C This subroutine creates the Scatter Matrix [XXT]
C *********************************************************************
C The subroutine creates the diagonal and lower diagonal elements
C of XXT in the loop structure and uses symmetry to create the
C upper diagonal elements to speed up the subroutine
C ***********INPUT VARIABLES*******************************************
C   XTILDE : Centred Trajectory Matrix
C     ITAU : Number of rows in Trajectory Matrix (tau)
C    ICOLS : Number of columns Trajectory Matrix (n=N-tau+1)
C ***********OUTPUT VARIABLES******************************************
C      XXT : Scatter Matrix XtildeXtilde'
C *********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)
      DIMENSION XTILDE(IMAXTAU,IMAXCOLS),XXT(IMAXTAU,IMAXTAU)

      DO I=1,ITAU
       DO J=I,ITAU
        XXT(I,J)=0.0D0
        DO II=1,ICOLS
         XXT(I,J)=XXT(I,J)+XTILDE(I,II)*XTILDE(J,II)
        ENDDO
        XXT(J,I)=XXT(I,J)
       ENDDO
      ENDDO
      RETURN
      END
```

```
C *********************************************************************
C                   PROJECTMAT [J. DE KLERK, 1999]
C *********************************************************************
      SUBROUTINE PROJECTMAT(V,ITAU,IR,IEVEC,PV)
C *********************************************************************
C This subroutine creates the Projection Matrix PV
C *********************************************************************
C The subroutine creates the diagonal and lower diagonal elements
C of PV in the loop structure and uses symmetry to create the
C upper diagonal elements to speed up the subroutine
C Note that the routine does not necessarily use the leading
C eigenvectors. The vector IEVEC contains the indices of the
C eigenvectors to use
C ***********INPUT VARIABLES*******************************************
C        V : Matrix with eigenvectors of XXT as its column vectors
C     ITAU : Number of rows in Trajectory Matrix (tau)
C       IR : Number of columns Trajectory Matrix (n=N-tau+1)
C    IEVEC : Vector containing index values of eigenvectors to use
C ***********OUTPUT VARIABLES******************************************
C       PV : Projection Matrix PV=VV'
C *********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)
      DIMENSION V(IMAXTAU,IMAXTAU),PV(IMAXTAU,IMAXTAU),IEVEC(40)

      DO I=1,ITAU
       DO J=I,ITAU
        PV(I,J)=0.0D0
        DO II=1,IR
         PV(I,J)=PV(I,J)+V(I,IEVEC(II))*V(J,IEVEC(II))
        ENDDO
        PV(J,I)=PV(I,J)
       ENDDO
      ENDDO
      RETURN
      END

C *********************************************************************
C                  CADZOWSIGNAL [J. DE KLERK, 2001]
C *********************************************************************
      SUBROUTINE CADZOWSIGNAL(F,INOBS,ITAU,IR,IEVEC,ICENT,SIGNAL,XMEAN,
     +PV,ITERATE)
C *********************************************************************
C This subroutine reconstructs a Cadzow signal series
C Note that this specific routine uses the eigenvectors that has
C indices provided in vector IEVEC
C ***********INPUT VARIABLES*******************************************
C        F : Vector of Time Series Observations
C    INOBS : Length of original series
C      IN1 : Length of series used for training series
C     ITAU : Number of rows in Trajectory Matrix (tau)
C       IR : Number of eigenvectors in base of Vr
C    IEVEC : Vector containing index values of eigenvectors to use
C ***********OUTPUT VARIABLES******************************************
C   SIGNAL : smoothed series (reconstructed signal series)
C    XMEAN : shift vector of r-flat L
C       PV : projection matrix of r-flat L
C  ITERATE : number of iterations until convergence
C *********************************************************************
```

```
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      DIMENSION F(IMAXOBS),SIGNAL(IMAXOBS)
      DIMENSION X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION XXT(IMAXTAU,IMAXTAU),V(IMAXTAU,IMAXTAU),EVAL(IMAXTAU)
      DIMENSION PV(IMAXTAU,IMAXTAU),XTILDE(IMAXTAU,IMAXCOLS)
      DIMENSION VEC1(IMAXTAU),VEC2(IMAXTAU),XHAT(IMAXTAU,IMAXCOLS)
      DIMENSION IEVEC(40)

C Open file 'C:\SHONG.DAT' which contains % variation explained
      OPEN(UNIT=4,FILE='C:\SHONG.DAT')

C Step 1 : Store the original time series as the signal series
      DO I=1,INOBS
       SIGNAL(I)=F(I)
      ENDDO
C Construct the signal series iteratively
C Project the column vectors of XTILDE onto linear subspace Vr
C and shift back to r-flat L. Average over reverse diagonals
C Iterate until convergence
      EPSILON=1.0D0
      ITERATE=0
  15  IF (EPSILON.GE.1E-9) THEN
C Set the number of columns in the trajectory matrix
       ICOLS=INOBS-ITAU+1
C Create Trajectory Matrix, X
       CALL TMAT(SIGNAL,ITAU,ICOLS,X)
C Create Centred Trajectory Matrix, XTILDE
       CALL CTMAT(ITAU,ICOLS,X,XMEAN,ICENT,XTILDE)
C Create Scatter Matrix, XXT
       CALL SCATMAT(XTILDE,ITAU,ICOLS,XXT)
C Calculate Eigenvectors of Matrix XXT that forms
C orthonormal basis for subspace Vr (IMSL routine)
       CALL DEVCSF(ITAU,XXT,IMAXTAU,EVAL,V,IMAXTAU)
C Determine the rank of matrix XXT, i.e. non-zero eigenvalues
C if rank is less than IR (eigenvectors requested) adjust IR
       IRANK=0
       DO IIR=1,IR
        IF (EVAL(IIR).GT.1E-9) THEN
         IRANK=IRANK+1
        ENDIF
       ENDDO
       ICONST=0
C Make provision for a series which is a constant
       IF (IRANK.LT.1) THEN
        IRANK=1
        ICONST=1
       ENDIF
       IF (IRANK.GT.IR) THEN
        IRANK=IR
       ENDIF
C Create Projection Matrix PV of linear subspace Vr
       CALL PROJECTMAT(V,ITAU,IRANK,IEVEC,PV)
C Exit if a constant model is found
       IF (ICONST.EQ.1) THEN
        RETURN
       ENDIF
```

```
C Calculate measure to check convergence (distance measure**2)
C Based on Frobenius norm
       SUM1=0.0D0
       DO I=IR+1,ITAU
        SUM1=SUM1+EVAL(I)
       ENDDO
       SUM2=0.0D0
       DO I=1,IR
        SUM2=SUM2+EVAL(I)
       ENDDO
       EPSILON=SUM1/SUM2

       IF (EPSILON.GE.1E-9) THEN
        ITERATE=ITERATE+1
C Create a new matrix XHAT by projecting the column vectors of
C matrix XTILDE onto Vr and shifting to r-flat L
        DO J=1,ICOLS
         DO I=1,ITAU
          SUM=0.0D0
          DO IJ=1,ITAU
           SUM=SUM+PV(I,IJ)*XTILDE(IJ,J)
          ENDDO
          XHAT(I,J)=SUM+XMEAN(I)
         ENDDO
        ENDDO

C Average over the reverse diagonals of matrix XHAT
C See algorithms in Danilov p.114
C Reconstruct f(i+1),...,f(i+tau)
        DO I=1,ITAU
         SIGNAL(I)=0.0D0
         DO J=1,I
          SIGNAL(I)=SIGNAL(I)+XHAT(J,I-J+1)/I
         ENDDO
        ENDDO
C Reconstruct f(i+tau+1),...,f(i+n)
         DO I=ITAU+1,ICOLS
          SIGNAL(I)=0.0D0
          DO J=1,ITAU
           SIGNAL(I)=SIGNAL(I)+XHAT(J,I-J+1)/ITAU
          ENDDO
         ENDDO
C Reconstruct f(i+n+1),...,f(i+N)
         DO I=ICOLS+1,INOBS
          SIGNAL(I)=0.0D0
          DO J=1,(INOBS-I+1)
           SIGNAL(I)=SIGNAL(I)+XHAT(J+I-ICOLS,ICOLS-J+1)/(INOBS-I+1)
          ENDDO
         ENDDO
        ENDIF
        GO TO 15
       ENDIF
C Calculate % variation explained by eigenvectors
       SUM1=0.0D0
       DO II=1,ITAU
        SUM1=SUM1+EVAL(II)
       ENDDO
       DO II=1,ITAU
        WRITE(4,'(G30.15)')EVAL(II)/SUM1
       ENDDO
       CLOSE(4)
      RETURN
      END
```

```
C ********************************************************************
C                    APPROXSIGNAL [J. DE KLERK, 2001]
C ********************************************************************
      SUBROUTINE APPROXSIGNAL(F,INOBS,ITAU,IR,IEVEC,ICENT,SIGNAL,XMEAN
     +,PV)
C ********************************************************************
C This subroutine constructs the approximate signal series. This is
C equivalent to using a single iteration in the Cadzow-signal series
C routine
C Note that the indices of eigenvectors that are used is supplied in
C the vector IEVEC
C ***********INPUT VARIABLES******************************************
C         F : Vector of Time Series Observations
C        IN : Time series length
C      ITAU : Number of rows in Trajectory Matrix (tau)
C        IR : Number of eigenvectors in base of Vr
C     IEVEC : Vector containing index values of eigenvectors to use
C **********OUTPUT VARIABLES******************************************
C    SIGNAL : smoothed series (reconstructed signal series)
C     XMEAN : shift vector of r-flat L
C        PV : projection matrix of r-flat L
C   ITERATE : number of iterations until convergence
C ********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      DIMENSION F(IMAXOBS),SIGNAL(IMAXOBS)
      DIMENSION X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION XXT(IMAXTAU,IMAXTAU),V(IMAXTAU,IMAXTAU),EVAL(IMAXTAU)
      DIMENSION PV(IMAXTAU,IMAXTAU),XTILDE(IMAXTAU,IMAXCOLS)
      DIMENSION VEC1(IMAXTAU),VEC2(IMAXTAU),XHAT(IMAXTAU,IMAXCOLS)
      DIMENSION IEVEC(40)

C Open file 'C:\SHONG.DAT' which contains % variation explained
      OPEN(UNIT=4,FILE='C:\SHONG.DAT')

C Set the number of columns in the trajectory matrix
      ICOLS=INOBS-ITAU+1
C Create Trajectory Matrix, X
      CALL TMAT(F,ITAU,ICOLS,X)
C Create Centred Trajectory Matrix, XTILDE
      CALL CTMAT(ITAU,ICOLS,X,XMEAN,ICENT,XTILDE)
C Create Scatter Matrix, XXT
      CALL SCATMAT(XTILDE,ITAU,ICOLS,XXT)
C Calculate Eigenvectors of Matrix XXT that forms
C orthonormal basis for subspace Vr (IMSL routine)
      CALL DEVCSF(ITAU,XXT,IMAXTAU,EVAL,V,IMAXTAU)
C Calculate % variation explained by eigenvectors
      SUM1=0.0D0
      DO II=1,ITAU
       SUM1=SUM1+EVAL(II)
      ENDDO
      DO II=1,ITAU
       WRITE(4,'(G30.15)')EVAL(II)/SUM1
      ENDDO
      CLOSE(4)

C Determine the rank of matrix XXT, i.e. non-zero eigenvalues
C if rank is less than IR (eigenvectors requested) adjust IR
      IRANK=0
      DO IIR=1,IR
       IF (EVAL(IIR).GT.1E-9) THEN
        IRANK=IRANK+1
       ENDIF
      ENDDO
      ICONST=0
C Make provision for a series which is a constant
      IF (IRANK.LT.1) THEN
       IRANK=1
       ICONST=1
      ENDIF
      IF (IRANK.GT.IR) THEN
       IRANK=IR
      ENDIF
C Create Projection Matrix PV of linear subspace Vr
      CALL PROJECTMAT(V,ITAU,IRANK,IEVEC,PV)

C Create a new matrix XHAT by projecting the column vectors of
C matrix XTILDE onto Vr and shifting to r-flat L
      DO J=1,ICOLS
       DO I=1,ITAU
        SUM=0.0D0
        DO IJ=1,ITAU
         SUM=SUM+PV(I,IJ)*XTILDE(IJ,J)
        ENDDO
        XHAT(I,J)=SUM+XMEAN(I)
       ENDDO
      ENDDO

C Hankelization operation
C Average over the reverse diagonals of matrix XHAT
C See algorithms in Danilov p.114
C Reconstruct f(i+1),...,f(i+tau)
      DO I=1,ITAU
       SIGNAL(I)=0.0D0
       DO J=1,I
        SIGNAL(I)=SIGNAL(I)+XHAT(J,I-J+1)/I
       ENDDO
      ENDDO
C Reconstruct f(i+tau+1),...,f(i+n)
      DO I=ITAU+1,ICOLS
       SIGNAL(I)=0.0D0
       DO J=1,ITAU
        SIGNAL(I)=SIGNAL(I)+XHAT(J,I-J+1)/ITAU
       ENDDO
      ENDDO
C Reconstruct f(i+n+1),...,f(i+N)
      DO I=ICOLS+1,INOBS
       SIGNAL(I)=0.0D0
       DO J=1,(INOBS-I+1)
        SIGNAL(I)=SIGNAL(I)+XHAT(J+I-ICOLS,ICOLS-J+1)/(INOBS-I+1)
       ENDDO
      ENDDO
      RETURN
      END
```

```
C **********************************************************************
C                     TRAJECTORYMAT [J. DE KLERK, 1999]
C **********************************************************************
      SUBROUTINE TRAJECTORYMAT(F,ITAU,ICOLS,X)
C **********************************************************************
C This subroutine creates the Trajectory Matrix [X]
C ***********INPUT VARIABLES********************************************
C        F : Vector of Time Series Observations
C     ITAU : Number of rows in Trajectory Matrix (tau)
C    ICOLS : Number of columns Trajectory Matrix (n=N-tau+1)
C ***********OUTPUT VARIABLES*******************************************
C        X : Trajectory Matrix
C **********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)
      DIMENSION F(IMAXOBS),X(IMAXTAU,IMAXCOLS)

      DO 2 I=1,ITAU
       DO 1 J=1,ICOLS
        X(I,J)=F(I+J-1)
    1  CONTINUE
    2 CONTINUE
      RETURN
      END

C **********************************************************************
C                    CTRAJECTORYMAT [J. DE KLERK, 1999]
C **********************************************************************
      SUBROUTINE CTRAJECTORYMAT(ITAU,ICOLS,X,XMEAN,XTILDE)
C **********************************************************************
C This subroutine creates the Centred Trajectory Matrix [XTILDE]
C ***********INPUT VARIABLES********************************************
C     ITAU : Number of rows in Trajectory Matrix (tau)
C    ICOLS : Number of columns Trajectory Matrix (n=N-tau+1)
C        X : Trajectory Matrix
C ***********OUTPUT VARIABLES*******************************************
C    XMEAN : Vector of matrix X row means (shift vector)
C   XTILDE : Centred Trajectory Matrix
C **********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)
      DIMENSION X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION XTILDE(IMAXTAU,IMAXCOLS)
      DO 3 I=1,ITAU
       XMEAN(I)=0.0D0
       DO 1 J=1,ICOLS
        XMEAN(I)=XMEAN(I)+X(I,J)/DFLOAT(ICOLS)
    1  ENDDO
       DO 2 J=1,ICOLS
        XTILDE(I,J)=X(I,J)-XMEAN(I)
    2  CONTINUE
    3 CONTINUE
      RETURN
      END
```

```
C **********************************************************************
C                     PROJECTMAT2 [J. DE KLERK, 1999]
C **********************************************************************
      SUBROUTINE PROJECTMAT2(V,ITAU,IR,PV)
C **********************************************************************
C This subroutine creates the Projection Matrix PV
C **********************************************************************
C The subroutine creates the diagonal and lower diagonal elements
C of PV in the loop structure and uses symmetry to create the
C upper diagonal elements to speed up the subroutine
C This subroutine uses leading eigenvectors to set up the projection
C matrix
C ***********INPUT VARIABLES********************************************
C        V : Matrix with eigenvectors of XXT as its column vectors
C     ITAU : Number of rows in Trajectory Matrix (tau)
C       IR : Number of columns Trajectory Matrix (n=N-tau+1)
C ***********OUTPUT VARIABLES*******************************************
C       PV : Projection Matrix PV=VV'
C **********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)
      DIMENSION V(IMAXTAU,IMAXTAU),PV(IMAXTAU,IMAXTAU)

      DO 3 I=1,ITAU
       DO 2 J=I,ITAU
        SUM=0.0D0
        DO 1 II=1,IR
         SUM=SUM+V(I,II)*V(J,II)
    1   CONTINUE
        PV(I,J)=SUM
        PV(J,I)=PV(I,J)
    2  CONTINUE
    3 CONTINUE
      RETURN
      END

C **********************************************************************
C                    FORECAST [J. DE KLERK, 2002]
C **********************************************************************
          SUBROUTINE FORECAST(F,IN1,IN2,ITAU,IR,IK,MAD,SSE,MSE,MAPE)
C **********************************************************************
C This subroutine forecasts a time series using SSA
C The forecasts are rolling 1-period-ahead forecasts
C ***********INPUT VARIABLES********************************************
C        F : Vector containing original time series
C      IN1 : Length of training time series
C      IN2 : Length of validation time series
C      TAU : Window length
C       IR : Number of leading eigenvectors
C       IK : Period-ahead forecasting
C       IR : Model selected using Forward Validation (FV)
C ***********OUTPUT VARIABLES*******************************************
C      MAD : Mean Absolute Deviation
C      SSE : Sum of Squared Errors
C      MSE : Mean Squared Error
C     MAPE : Mean Absolute Deviation
C ***********OTHER VARIABLES*******************************************
C        X : Trajectory Matrix
```

```
C   XTILDE : Centred Trajectory Matrix
C      XXT : Scatter matrix of matrix XTILDE
C        V : Matrix of eigenvectors of the scatter matrix XXT
C     EVAL : Vector of eigenvalues of the scatter matrix XXT
C       PV : Projection matrix of parallel linear subspace Vr
C        P : Projection matrix of orthogonal parallel linear subspace
C       P1 : Partitioning of matrix P
C       P2 : Partitioning of matrix P
C    P2TP2 : Matrix P2'*P2
C    P2TP1 : Matrix P2'*P1
C P2TP2INV : Matrix inv(P2'P2)
C    XMEAN : Shift vector to r-flat
C ********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      DIMENSION F(IMAXOBS),X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION XTILDE(IMAXTAU,IMAXCOLS),XXT(IMAXTAU,IMAXTAU)
      DIMENSION PV(IMAXTAU,IMAXTAU)
      DIMENSION P1(IMAXTAU,IMAXTAU),P2(IMAXTAU,IMAXTAU)
      DIMENSION P(IMAXTAU,IMAXTAU),P2TP2(200,200)
      DIMENSION P2TP1(IMAXTAU,IMAXTAU),P2TP2INV(200,200)
      DIMENSION P2TP2IP2TP1(IMAXTAU,IMAXTAU)
      DIMENSION VEC1(IMAXTAU),FHAT(IMAXOBS)
      DIMENSION EVAL(IMAXTAU),V(IMAXTAU,IMAXTAU)

C Rest measure of out-of-sample forecasting accuracy
      MAD=0.0D0
      SSE=0.0D0
      MSE=0.0D0
      MAPE=0.0D0
C Set position in original series to start at
      IN3=IN1-IK+1
      IN4=IN3
C Loop over the validation series observations
      DO 101 I1=1,IN2
       ICOLS=IN3-ITAU+1
C Create Trajectory Matrix, X
       CALL TRAJECTORYMAT(F,ITAU,ICOLS,X)
C Create Centred Trajectory Matrix, XTILDE
       CALL CTRAJECTORYMAT(ITAU,ICOLS,X,XMEAN,XTILDE)
C Create Scatter Matrix, XXT
       CALL SCATMAT(XTILDE,ITAU,ICOLS,XXT)
C Calculate Eigenvectors of Matrix XXT that forms
C orthonormal basis for subspace Vr (IMSL routine)
       CALL DEVCSF(ITAU,XXT,IMAXTAU,EVAL,V,IMAXTAU)
C Determine the rank of matrix XXT, i.e. non-zero eigenvalues
C if rank is less than IR (eigenvectors requested) adjust IR
       IRANK=0
       DO IIR=1,IR
        IF (EVAL(IIR).GT.1E-9) THEN
         IRANK=IRANK+1
        ENDIF
       ENDDO
C Make provision for a series which is a constant
       IF (IRANK.LT.1) THEN
        IRANK=1
       ENDIF
C Re-set number eigenvalues, if necessary. Make provision for perfect fit
       IF (IR.LT.IRANK) THEN
        IRANK=IR
       ENDIF

C Create Projection Matrix PV of linear subspace Vr
       CALL PROJECTMAT2(V,ITAU,IRANK,PV)

C Create projection matrix of orthogonal parallel linear subspace Vr
       DO I=1,ITAU
        DO J=I,ITAU
         IF (I.EQ.J) THEN
          P(I,J)=1.0D0-PV(I,J)
         ELSE
          P(I,J)=0.0D0-PV(I,J)
         ENDIF
         P(J,I)=P(I,J)
        ENDDO
       ENDDO

C Partition projection matrix P into P1 and P2
       DO I=1,ITAU
        DO J=1,ITAU
         IF (J.LE.(ITAU-IK)) THEN
          P1(I,J)=P(I,J)
         ENDIF
         IF (J.GE.(ITAU-IK)+1) THEN
          P2(I,J-(ITAU-IK))=P(I,J)
         ENDIF
        ENDDO
       ENDDO

C Create P2'P2 and invert the result
       DO I=1,IK
        DO J=1,IK
         P2TP2(I,J)=0.0D0
         DO IJ=1,ITAU
          P2TP2(I,J)=P2TP2(I,J)+P2(IJ,I)*P2(IJ,J)
         ENDDO
         P2TP2(J,I)=P2TP2(I,J)
        ENDDO
       ENDDO

C Set up inverse of P2TP2
       CALL DLINDS(IK,P2TP2,200,P2TP2INV,200)

C Create P2'P1 and invert the result
       DO I=1,IK
        DO J=1,(ITAU-IK)
         P2TP1(I,J)=0.0D0
         DO IJ=1,ITAU
          P2TP1(I,J)=P2TP1(I,J)+P2(IJ,I)*P1(IJ,J)
         ENDDO
        ENDDO
       ENDDO

C Create inv(P2'P2)P2TP1
       DO I=1,IK
        DO J=1,(ITAU-IK)
         P2TP2IP2TP1(I,J)=0.0D0
         DO IJ=1,IK
          P2TP2IP2TP1(I,J)=P2TP2IP2TP1(I,J)+P2TP2INV(I,IJ)*P2TP1(IJ,J)
         ENDDO
        ENDDO
       ENDDO
```

```
C Create k-step-ahead forecasts for the multiple time series
       ICOUNT=0
       DO II=1,ITAU-IK
        ICOUNT=ICOUNT+1
        VEC1(ICOUNT)=F(IN3-(ITAU-IK)+II)-
     +  XMEAN(ICOUNT)
       ENDDO

       DO I=1,IK
        FHAT(I)=0.0D0
        DO J=1,(ITAU-IK)
         FHAT(I)=FHAT(I)+P2TP2IP2TP1(I,J)*VEC1(J)
        ENDDO
        FHAT(I)=XMEAN((ITAU-IK)+I)-FHAT(I)
       ENDDO

C Calculate measures of out-of-sample forecasting accuracy
C based only on k-th forecast
       MAD=MAD+DABS(F(IN4+IK)-FHAT(IK))/
     + DFLOAT(IN2)
       SSE=SSE+((F(IN4+IK)-FHAT(IK))**2)
       MSE=MSE+((F(IN4+IK)-FHAT(IK))**2)/
     + DFLOAT(IN2)
       MAPE=MAPE+DABS(F(IN4+IK)-FHAT(IK))/
     + (DFLOAT(IN2)*DABS(F(IN4+IK)))*100

C Increase training series length by 1
        IN3=IN3+1
        IN4=IN4+1
  101  CONTINUE
      END

C ******************************************************************
C             MEASURES [J.DE KLERK, 1999]
C ******************************************************************
      SUBROUTINE MEASURES(INOBS,F,FHAT,MAD,MAPE,SSE,MSE)
C ******************************************************************
C This subroutine calculates the measures of forecasting accuracy
C ***********INPUT VARIABLES******************************************
C        F : Time series vector
C     FHAT : Vector of Predictions
C    INOBS : Length of vectors
C ***********OUTPUT VARIABLES*****************************************
C      MAD : Mean Absolute Deviation
C     MAPE : Mean Absolute Percentage Error
C      SSE : Sum of Squared Errors
C      MSE : Mean Square Error
C ***************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
      DIMENSION F(IMAXOBS),FHAT(IMAXOBS)

      SSE=0.0D0
      MAD=0.0D0
      MAPE=0.0D0

      DO I=1,INOBS
       SSE=SSE+(F(I)-FHAT(I))**2
       MAD=MAD+(ABS(F(I)-FHAT(I)))/DFLOAT(INOBS)
       IF (ABS(F(I)).GT.0) THEN
        MAPE=MAPE+(ABS(F(I)-FHAT(I))/F(I))
       ELSE
        MAPE=MAPE+0.0D0
       ENDIF
      ENDDO

      IF (SSE.LE.1E-10) THEN
       SSE=0.0D0
      ENDIF

      IF (MAD.LE.1E-10) THEN
       MAD=0.0D0
      ENDIF

      IF (MAPE.GE.1E-10) THEN
       MAPE=MAPE/DFLOAT(INOBS)*100
      ELSE
       MAPE=0.0D0
      ENDIF

      MSE=SSE/INOBS
      RETURN
      END

C ********************************************************************
C                  SIGNALSERIES [J. DE KLERK, 2001]
C ********************************************************************
      SUBROUTINE SIGNALSERIES(F,INOBS,IN1,ITAU,ICOUNTR,SIGNAL,
     +XMEAN,PV,ITERATE,ISERIES)
C ********************************************************************
C This subroutine returns a signal series with its shift vector and
C projection matrix of the subspace
C ***********INPUT VARIABLES******************************************
C        F : Vector of Time Series Observations
C    INOBS : Length of time series
C      IN1 : Length of training series
C     ITAU : Window length
C  ICOUNTR : Number of eigenvectors in "model"
C  ISERIES : If ISERIES=0 No iteration are performed and the signal
C            series returned is the original series with PV and XMEAN
C          : If ISERIES=1 A single iteration is performed and the
C            approximate series with PV and XMEAN is returned
C          : If ISERIES=2 The Cadzow-signal is constructed and PV,
C            XMEAN is returned
C ***********OUTPUT VARIABLES*****************************************
C   SIGNAL : smoothed series (reconstructed signal series)
C    XMEAN : shift vector of r-flat L
C       PV : projection matrix of r-flat L
C  ITERATE : number of iterations until convergence
C ********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,K-Z),INTEGER(I-J)
C Set maximum time series length
      PARAMETER (IMAXOBS=1000)
C Set maximum window length (tau) automatically
      PARAMETER (IMAXTAU=INT((IMAXOBS+1)/2))
C Set maximum number of columns in trajectory matrix automatically
      PARAMETER (IMAXCOLS=IMAXOBS-1)

      DIMENSION F(IMAXOBS),SIGNAL(IMAXOBS)
      DIMENSION X(IMAXTAU,IMAXCOLS),XMEAN(IMAXTAU)
      DIMENSION XXT(IMAXTAU,IMAXTAU),V(IMAXTAU,IMAXTAU),EVAL(IMAXTAU)
      DIMENSION PV(IMAXTAU,IMAXTAU),XTILDE(IMAXTAU,IMAXCOLS)
      DIMENSION VEC1(IMAXTAU),VEC2(IMAXTAU),XHAT(IMAXTAU,IMAXCOLS)
      DIMENSION XHAT1(IMAXTAU,IMAXCOLS),IPC(40)

C Set the number of eigenvectors to use
```

```
      IR=ICOUNTR
C Step 1 : Store the original time series as the signal series
C Note : Use only the first N1 elements of the series
      DO I=1,INOBS
       SIGNAL(I)=F(I)
      ENDDO
C Construct the signal series iteratively
C Project the column vectors of XTILDE onto linear subspace Lr
C and shift back to r-flat L. Average over reverse diagonals
C Iterate until convergence
      EPSILON=1.0D0
      ITERATE=0
  15  IF (EPSILON.GE.1E-6) THEN
C Set the number of columns in the trajectory matrix
       ICOLS=IN1-ITAU+1
C Create Trajectory Matrix, X
       CALL TRAJECTORYMAT(SIGNAL,ITAU,ICOLS,X)
C Create Centred Trajectory Matrix, XTILDE
       CALL CTRAJECTORYMAT(ITAU,ICOLS,X,XMEAN,XTILDE)
C Create Scatter Matrix, XXT
       CALL SCATMAT(XTILDE,ITAU,ICOLS,XXT)
C Calculate Eigenvectors of Matrix XXT that forms
C orthonormal basis for subspace Vr (IMSL routine)
       CALL DEVCSF(ITAU,XXT,IMAXTAU,EVAL,V,IMAXTAU)
C Create Projection Matrix PV of linear subspace Vr
       CALL PROJECTMAT(V,ITAU,IR,IPC,PV)

C Calculate measure to check convergence (distance measure**2)
       SUM1=0.0D0
       DO IK=1,IR
        SUM1=SUM1+EVAL(IK)
       ENDDO
       SUM2=0.0D0
       DO IK=IR+1,ITAU
        SUM2=SUM2+EVAL(IK)
       ENDDO
       EPSILON=SUM2/SUM1

C The following statement is used to sidestep the signal reconstructive part if
C if the original series must be used
       IF (ISERIES.EQ.0) THEN
        EPSILON=1E-8
       ENDIF
C The following statement is used to perform a single Cadzow iteration
       IF ((ISERIES.EQ.1).AND.(ITERATE.EQ.1)) THEN
        EPSILON=1E-8
       ENDIF

       IF (EPSILON.GE.1E-6) THEN
        ITERATE=ITERATE+1
C Create a new matrix XHAT by projecting the column vectors of
C matrix XTILDE onto Vr and shifting to r-flat L
        DO J=1,ICOLS
         DO I=1,ITAU
          SUM=0.0D0
          DO IJ=1,ITAU
           SUM=SUM+PV(I,IJ)*XTILDE(IJ,J)
          ENDDO
          XHAT(I,J)=SUM+XMEAN(I)
         ENDDO
        ENDDO

C Average over the reverse diagonals of matrix XHAT
C See algorithms in Danilov p.12
C Reconstruct f(1),...,f(tau)
        DO I=1,ITAU
         SIGNAL(I)=0.0D0
         DO J=1,I
          SIGNAL(I)=SIGNAL(I)+XHAT(J,I-J+1)/I
         ENDDO
        ENDDO
C Reconstruct f(tau+1),...,f(n)
        DO I=ITAU+1,ICOLS
         SIGNAL(I)=0.0D0
         DO J=1,ITAU
          SIGNAL(I)=SIGNAL(I)+XHAT(J,I-J+1)/ITAU
         ENDDO
        ENDDO
C Reconstruct f(n+1),...,f(N)
        DO I=ICOLS+1,IN1
         SIGNAL(I)=0.0D0
         DO J=1,(IN1-I+1)
          SIGNAL(I)=SIGNAL(I)+XHAT(J+I-ICOLS,ICOLS-J+1)/(IN1-I+1)
         ENDDO
        ENDDO
       ENDIF
       GO TO 15
      ENDIF
      RETURN
      END
```

REFERENCES

Allen, M.R. and Smith, L.A. (1996) Monte Carlo SSA: Detecting Irregular Oscillations in the Presence of Colored Noise. *Journal of Climate*, **9**, 3373-3404.

Bancoff, J.W. (1983) *Linear Algebra Through Geometry*. New York: Springer-Verlag.

Becker, C. and Gather, U. (1999) The Masking Breakdown Point of Multivariate Outlier Identification Rules. *Journal of the American Statistical Association*, **94**, 947-955.

Bowerman, B.L. and O'Connell, R.T. (1993) *Forecasting and Time Series: An Applied Approach*. Duxbury Press.

Box, G.E.P. and Jenkins, G.M. (1970) *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day.

Bragg, M. (1998) *On Giant's Shoulders – Great Scientists and their Discoveries from Archimedes to DNA*. Sceptre Publishing.

Brock, W.A., Dechert, W.D., Scheinkman, J.A. and LeBaron, B. (1996) A Test for Independence Based on the Correlation Dimension. *Econometric Reviews*, **15**, 197-235.

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

Buchstaber, V.M. (1994) Time Series Analysis and Grassmannians. *Amer. Math. Soc. Transl.*, **162**, 1-17.

Burnham, K.P. and Anderson, D.R. (1998) *Model Selection and Inference*. New York: Springer-Verlag.

Business Forecast Systems (1999) *Forecast Pro for Windows*. Belmont: BFS.

Cadzow, J.A. (1988) Signal Enhancement - A Composite Property Mapping Algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **36**, 49-62.

Chatfield, C. (1993) Calculating Interval Forecasts. *Journal of Business & Economic Statistics*, **11**, 121-135.

Chatfield, C. (1994) *The Analysis of Time Series – An Introduction*. London:Chapman and Hall.

Croux, C. and Haesbroeck, G. (2000) Principal Component Analysis based on Robust Estimators of the Covariance or Correlation Matrix: Influence functions and efficiencies. *Biometrika*, **87**, 603-618.

Danilov, D.L. (1997) Principal Components in Time Series Forecast. *Journal of Computational and Graphical Statistics*, **6**, 112-121.

Davison, A.C. and Hinkley, D.V. (1999) *Bootstrap Methods and their Applications*. Cambridge University Press.

Devlin, S.J., Gnanadesikan, R and Kettenring, J.R. (1981) Robust Estimation of Dispersion Matrices and Principal Components. *Journal of the American Statistical Association*, **76**, 354-362.

Diebold, F.X. (1998) *Elements of Forecasting*. Cincinnati: South-Wester College Publishing.

Efron, B. and Tibshirani, R.J. (1998) *An Introduction to the Bootstrap*. New York: Chapman & Hall/CRC.

Elsner, J.B. and Tsonis, A.A. (1996) *Singular Spectum Analysis: A New Tool in Time Series Analysis*. New York: Plenum Press.

Farraway, J. and Chatfield, C. (1998) Time Series forecasting with neural networks: a comparative study using airline data. *Applied Statist.*, **47**, 231-250.

Flury, B.D. (1997) *A First Course in Multivariate Statistics*. New York: Springer-Verlag.

Fowler, A.C. and Kember, G. (1998) Singular system analysis as a moving-window spectral method. *European Journal of Applied Mathematics*, **9**, 55-79.

Fraleigh, J.B. and Beauregard, R.A. (1987) *Linear Algebra*. Massachusetts: Addison-Wesley Publishing Company.

Gentle, J.E. (1998) *Numerical Linear Algebra for Application in Statistics*. New York: Springer.

Ghil, M. and Vautard, R. (1991) Interdecadal oscillations and the warming trend in global temperature time series, *Nature*, **350**, 324-327.

Gill, P.E, Murray, W., Wright, M.H. (1991) *Numerical Linear Algebra and Optimization*. California: Addison-Wesley Publishing Company.

Gleick, J. (1987) *CHAOS Making a New Science*. Viking.

Golyandina, N., Nekrutkin, V., Zhigljavsky, A. (2001) *Analysis of Time Series Structure – SSA and Related Techniques. Monographs on Statistics and applied Probability*, **90**, Chapman and Hall/CRC.

Gower, J.C. and Hand, D.J. (1996) *Biplots*. New-York: Chapman & Hall.

Granger, C.W.J. and Newbold, P. (1986) *Forecasting Economic Time Series*. New York: Academic Press.

Harvey, A.C. (1993) *Time Series Models*. MIT Press.

Harville, D.A. (1997) *Matrix Algebra from a Statistician's Perspective*. New York: Springer-Verlag.

Hjorth, U. (1999) *Computer Intensive Statistical Methods*. London: Chapman and Hall.

Huber, P.J. (1981) *Robust Statistics*. New York: John Wiley & Sons, Inc.

Hurt, N.E. (1991) Signal enhancement and the method of successive projections. *Acta Applicandae Mathematicae*, **23**, 145-162.

Jackson, J.E. (1991) *A User's Guide to Principal Components*. New York: Springer-Verlag. pp. 365-371.

Jolliffe, I.T. (1986) *Principal Component Analysis*. New-York: Springer-Verlag.

Kamiya, H. and Eguchi, S. (2001) A Class of Robust Principal Component Vectors. *Journal of Multivariate Analysis*, **77**, 239-269.

Kosinski, A.S. (1999) A procedure for the detection of multivariate outliers. *Computational Statistics and Data Analysis*, **29**, 145-161.

Krzanowski, W.J. (2000) *Principles of Multivariate Analysis: A User's Perspective*. Oxford University Press.

Kuan, C.-M. and Liu, T. (1995) Forecasting Exchange Rates using Feedforward and Recurrent Neural Networks. *Journal of Applied Econometrics*, **10**, 347-364.

LeBaron, B. (1997) A Fast Algorithm for the BDS Statistic. *Studies in Nonlinear Dynamics and Econometrics*, **2**, 53-59.

Liebert, W., and Schuster, H.G. (1988) Proper Choice of the Time Delay for the Analysis of Chaotic Time Series. *Physical Letters A*, **142**, 107-111.

Locantore, N., Maronna, J.S., Simpson, D.G., Tripoli, N., Zhang, J.T. and Cohen, K.L. (1999) Robust principal component analysis for functional data. *Test*, **8**, 1-73.

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E. and Winkler, R. (1982) The Accuracy of Extrapolation (Time Series) Methods: Results of a Forecasting Competition. *Journal of Forecasting*, **1**, 111-153.

Makridakis, S., Wheelwright, S.C. and Hyndman, R. (1998) *Forecasting: Methods and Applications*. New York: John Wiley & Sons, Inc.

Marden, J.I. (1999) Some robust estimates of principal components. *Statistics and Probability Letters*, **43**, 349-359.

Maronna, R.A. and Yohai, V.J. (1998) Robust estimation of multivariate location and scatter. In *Encyclopedia of Statistical Science Update*, 2, Ed. S. Kotz, C. Read and D. Banks, pp.589-596. New York: Wiley.

McQuarrie, A.D.R. and Tsai, C.L. (1998) *Regression and Time Series Model Selection*. London: World Scientific Publishing.

Morrison, D.F. (1990) *Multivariate Statistical Methods*. New York: McGraw-Hill Publishing Company.

Ord, K. and Lowe, S. (1996) Automatic Forecasting. *The American Statistician*, **50**, 88-94.

Packard, N.H., Crutchfield, J.P., Farmer, J.D. and Shaw, R.S. (1980) Geometry from a Time Series. *Physical Review Letters*, **45**, 712-716.

Paluš, M. and Nvotná, D. (1998) Detecting modes with nontrivial dynamics embedded in colored noise: enhanced Monte Carlo SSA and the case of climate oscillations. *Physical Letters A*, **248**, 191-202.

Park, H., Zhang, L. and Rosen, J.B. (1999) Low rank approximation of a Hankel matrix by structured total least norm. *BIT*, **39**, 757-779.

Peña, D., Tiao, G.C. and Tsay, R.S. (2001) *A Course in Time Series Analysis*. New York: John Wiley & Sons, Inc.

Peters, E.E. (1991) *Chaos and Order in The Capital Markets*. New York: John Wiley & Sons, Inc.

Plaut, G. and Vautard, R. (1994) Spells of Low-Frequency and Weather Regimes in the Northern Hemisphere. *Journal of the Atmospheric Sciences*, **51**, 210-236.

Rasband, S.N. (1990) *Chaotic Dynamics of Nonlinear Systems*. New York: John Wiley.

Quantitative Micro Software (2000) EVIEWS4 User's Guide.

Rousseeuw, P.J and Van Zomeren, B.C. (1990) Unmasking Multivariate Outliers and Leverage Points. *Journal of the American Statistical Association* , **85**, 633-639.

Rousseeuw, P.J. and Van Driessen, K. (1999) A Fast Algorithm for the Minimum Covariance Determinant Estimator. *Technometrics*, **41**, 212-223.

Scott, D.W. (1992) *Multivariate Density Estimation: Theory, Practice and Visualization*. New York: John Wiley & Sons, Inc.

Shao, J. and Tu, D. (1995) *The Jackknife and Bootstrap*. New York: Springer-Verlag.

Shumway, R.H. and Stoffer, D.S. (2000) *Time Series Analysis and Its Applications*. New York: Springer-Verlag.

Small, C.G. (1990) A Survey of Multidimensional Medians. *International Statistical Review*, **58**, 263-277.

Stine, R.A. (1985) Bootstrap Prediction Intervals for Regression. *Journal of the American Statistical Association*, **80**, 1026-1031.

Stoica, P. and Moses, R.L. (1997) *Introduction to Spectral Analysis*. New Jersey: Prentice Hall.

Takens, F. (1981) Detecting strange attractors in turbulence. In D.A. Rand and L.-S. Young (Eds.), *Dynamical Systems and Turbulence*, volume 898 of *Lecture notes in Mathematics*, 366-381. Springer-Verlag, Berlin.

Thisted, R.A. (2000) *Elements of Statistical Computing*: *Numerical Computation*. New York: Chapman & Hall/CRC.

Trefethen, L.N. and Bau, D. III (1997) *Numerical Linear Algebra*. Philadelphia: SIAM.

Varadi, F., Pap, J.M., Ulrich, R.K., Bertello, L. and Henney, C.J. (1999) Searching for Signal in Noise by Random-Lag Singular Spectrum Analysis. *The Astrophysical Journal*, **526**, 1052-1061.

Vautard, R. and Ghil, M. (1989) Singular Spectrum Analysis in Nonlinear Dynamics, with Applications to Paleeclimatic Time Series. *Physica D*, **20**, 395-424.

Vautard, R., Yiou, P. and Ghil, M. (1992) Singular-spectrum analysis: A toolkit for short, noisy chaotic signals. *Physica D*, **58**, 95-126.

Venables, W.N. and Ripley, B.D. (1999) *Modern Applied Statistics with S-PLUS*. New York: Springer-Verlag.

Venter, J.H. (1998) *Forecasting by identification of linear structure in a time series*. Unpublished talk presented at the 1998 Conference of The South-African Statistical Association.

Weigend, A.S., Huberman, B.A. and Rumelhart, D.E. (1990) Predicting the Future: A Connectionist Approach. *International Journal of Neural Systems*, **1**, 193-209.

Wickens, T.D. (1995) *The Geometry of Multivariate Statistics*. New Jersey: Lawrence Erlbaum Associates, Publishers.

Wilks, S.S. (1963) Multivariate Statistical Outliers, *Sankhy*$\overline{a}$, **25**, 407-426.

Yong-Seok, C. and Myung-Hoe, H. (1996) Resistant Singular Value Decomposition and Its Statistical Applications. *Journal of the Korean Statistical Society*, **25**, 49-66.

Zhang, B., Lie, J., Sun, Z. (1993) A new multidimensional time series forecasting method based on the EOF iteration scheme. *Advances in Atmospheric Sciences*, **10**, 243-247.