The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.
Empirical Modelling of High-Frequency Foreign Exchange Rates

by

Someshini Packirisamy

B.Sc.(Hon.), University of Cape Town (1999)

Submitted to the Department of Mathematics and Applied Mathematics in partial fulfillment of the requirements for the degree of

Masters of Science in Mathematics of Finance

at the

UNIVERSITY OF CAPE TOWN

June 2004

©

The author hereby grants to University of Cape Town permission to reproduce and to distribute copies of this thesis document in whole or in part.

Signature of Author

Department of Mathematics and Applied Mathematics

16 February 2004

Certified by

Dr Renkuan Guo

Associate Professor, Department of Statistical Sciences, University of Cape Town

Thesis Supervisor

Accepted by

Professor B.D. Reddy

Dean, Faculty of Science, University of Cape Town
Empirical Modelling of High-Frequency Foreign Exchange Rates

by

Someshini Packirisamy

Submitted to the Department of Mathematics and Applied Mathematics
on 16 February 2004, in partial fulfillment of the
requirements for the degree of
Masters of Science in Mathematics of Finance

Abstract

Foreign exchange markets around the world trade twenty-four hours a day, for seven days a week. Time series data may be sampled from these markets at various intervals, with the highest frequency being tick-by-tick (i.e. as data reaches the market). High frequency data usually refers to time series data sampled, from financial markets, at intervals less than or equal to daily.

There is a wealth of information available on modelling foreign exchange time series data, however, research studies on modelling and predicting high frequency foreign exchange data is less prominent. Furthermore, there does not appear to be much evidence supporting work on the modeling and prediction of high frequency South African Rand/United States Dollar (ZAR/USD) exchange rates. A fair amount of noise is embedded in high frequency time series data, in particular for the ZAR/USD exchange rate, and the modelling of these time series requires the use of specialized models. In addition lengthy high frequency foreign exchange data is largely unavailable for the South African market.

This dissertation undertakes empirical explorations to model high frequency foreign exchange time series (primarily the ZAR/USD time series), through the use of multi-agent neural networks, linear Kalman filters and fuzzy Markov chain theory.

Thesis Supervisor: Dr Renkuan Guo
Title: Associate Professor, Department of Statistical Sciences, University of Cape Town
Contents

1 INTRODUCTION . 14
1.1 Subject of the Dissertation ........................................... 14
1.2 Objectives of the Dissertation ........................................ 14
1.3 Scope and Limitations of the Dissertation .......................... 15
1.4 Organisation of the Dissertation ..................................... 16
1.5 Softwares used in the Dissertation ................................... 17

2 LITERATURE REVIEW 19
2.1 Overview of the Foreign Exchange (Forex) Market ................ 19
2.2 Stylized Features of Financial Data .................................. 20
2.3 High Frequency Data .................................................... 22
2.4 Kalman Filters to Model Financial Time Series ................. 22
2.5 Neural Networks to Model Financial Time Series ............... 26
  2.5.1 Multi-Agent Modelling of FX-Markets by Neural Networks .. 26

3 EXPLORATIONS OF FOREX DATA USING TRADITIONAL STATISTICAL METHODS 30
3.1 Concepts of Statistical Time Series .................................. 30
3.2 Data Set ........................................................................ 32
3.3 Statistical Analysis of Daily ZAR/USD Time Series ............. 33
  3.3.1 Time Series Plots ...................................................... 33
3.3.2 Test for Stationarity ................................................... 35
3.3.3 Histogram and Descriptive Statistics ............................ 36
3.3.4 Normal Quantile-Quantile Plots ........................................ 38
3.3.5 Autocorrelation and Partial Correlation ............................ 39

3.4 Statistical Analysis of Daily DEM/USD Time Series ............... 41
3.4.1 Time Series Plots .................................................. 41
3.4.2 Test for Stationarity ............................................... 42
3.4.3 Histogram and Descriptive Statistics ................................ 43
3.4.4 Normal Quantile-Quantile Plots .................................. 44
3.4.5 Autocorrelation and Partial Autocorrelation ...................... 45

3.5 Statistical Analysis of 5 Minute ZAR/USD Time Series .......... 46
3.5.1 Time Series Plots .................................................. 46
3.5.2 Test for Stationarity ............................................... 48
3.5.3 Histogram and Descriptive Statistics ............................. 49
3.5.4 Normal Quantile-Quantile Plots .................................. 51
3.5.5 Autocorrelation and Partial Correlation ......................... 52

3.6 Statistical Analysis of 15 Minute ZAR/USD Time Series .......... 53
3.6.1 Time Series Plots .................................................. 53
3.6.2 Test for Stationarity ............................................... 55
3.6.3 Histogram and Descriptive Statistics ............................. 56
3.6.4 Normal Quantile-Quantile Plots .................................. 58
3.6.5 Autocorrelation and Partial Correlation ......................... 59

3.7 Statistical Analysis of 30 Minute ZAR/USD Time Series .......... 60
3.7.1 Time Series Plots .................................................. 60
3.7.2 Test for Stationarity ............................................... 62
3.7.3 Histogram and Descriptive Statistics ............................. 63
3.7.4 Normal Quantile-Quantile Plots .................................. 65
3.7.5 Autocorrelation and Partial Correlation ......................... 66

3.8 Statistical Analysis of 60 Minute ZAR/USD Time Series .......... 67
3.8.1 Time Series Plots .................................................. 67
3.8.2 Test for Stationarity ............................................... 69
3.8.3 Histogram and Descriptive Statistics ............................. 70
| 3.8.4 | Normal Quantile-Quantile Plots | 71 |
| 3.8.5 | Autocorrelation and Partial Correlation | 72 |
| 3.9 | Concluding Remarks | 74 |

## 4 FITTING A STABLE DISTRIBUTION AND R/S ANALYSIS OF FOREX DATA

4.1 Data Set ............................................................................................................. 75
4.2 Fitting a Stable Distribution to Foreign Exchange Data ...................................... 75
   4.2.1 Overview of Stable Distributions ................................................................... 75
   4.2.2 Fitting the Stable Distribution ..................................................................... 77
   4.2.3 Concluding Remarks .................................................................................... 83
4.3 R/S Analysis and the Hurst Exponent .................................................................. 84
   4.3.1 Overview of R/S Analysis and the Hurst Exponent ....................................... 84
   4.3.2 R/S Analysis of ZAR/USD Returns .............................................................. 86
   4.3.3 Concluding Remarks ..................................................................................... 91

## 5 MULTI-AGENT MODELLING OF SA FOREX MARKET BY NEURAL NETWORKS

5.1 An Introduction to Neural Networks ................................................................... 92
   5.1.1 The Benefits of Neural Networks .................................................................. 94
   5.1.2 Artificial Intelligence and Neural Networks .................................................. 96
5.2 Multi-Agent Modelling of a Foreign Exchange Market ....................................... 108
   5.2.1 Explicit Dynamics of a Foreign Exchange Market ....................................... 114
   5.2.2 SENN Analysis ......................................................................................... 126
   5.2.3 Discussion of Results ................................................................................. 130
   5.2.4 Other Neural Network Implementations .................................................... 135
   5.2.5 Concluding Remarks ................................................................................... 138

## 6 THE KALMAN FILTER IN FINANCE

6.1 Introduction to State-Space Models .................................................................... 140
   6.1.1 The Discrete Kalman Filter Algorithm ....................................................... 142
6.2 The State-Space Representation of a Dynamic System ........................................ 143
  6.2.1 Derivation of the Kalman Filter ................................................................. 144
  6.2.2 Forecasting with the Kalman filter ............................................................... 148
  6.2.3 Statistical Inference about Unknown Parameters using the Kalman filter ....... 150
  6.2.4 Smoothing ...................................................................................................... 153
  6.2.5 Time-Varying Coefficient Models .................................................................. 155
6.3 State-Space Modelling in EViews using the Kalman filter .................................. 157
  6.3.1 Modelling the DEM/USD Returns ................................................................. 157
  6.3.2 Modelling the ZAR/USD Returns ................................................................. 164
6.4 Concluding Remarks ............................................................................................ 172

7 FUZZY MARKOV CHAIN MODELLING OF HIGH FREQUENCY FOREX DATA .......... 176
  7.1 Fuzzy Uncertainty .............................................................................................. 176
  7.2 Fuzzy Probability Calculus ................................................................................ 179
    7.2.1 Probability of Fuzzy Events ....................................................................... 179
    7.2.2 Relative membership Degree ...................................................................... 182
    7.2.3 Sample Membership for ZAR/USD 5 minute data ...................................... 183
  7.3 Markov Chain with Fuzzy states ........................................................................ 185
    7.3.1 Markov Chains ............................................................................................. 185
    7.3.2 Likelihood theory for Markov chains ........................................................... 186
    7.3.3 Likelihood estimator for a Markov Chain with Fuzzy States ...................... 188
  7.4 Prediction using Markov chain theory .................................................................. 190
  7.5 The Estimator for the Observation $u_{m+1}$ ...................................................... 191
    7.5.1 Point Estimate of FX rate and its Confidence Interval ............................... 193
  7.6 Fuzzy Markov Chain Theory applied to ZAR/USD 5 Minute Observations ........ 193
    7.6.1 Overall ZAR/USD 5 Minute Data Set (01/07/2002 until 31/01/2003) .......... 193
    7.6.2 Partitioned ZAR/USD 5 minute data set ...................................................... 199
  7.7 Concluding Remarks .......................................................................................... 207

8 CONCLUDING REMARKS ......................................................................................... 209
A SENN CODES

A.1 SENN MODEL FILE .......................................................... 220
A.2 SENN TOPOLOGY FILE ......................................................... 230

B FUZZY MARKOV CHAIN MODELLING CALCULATIONS .............. 231

B.1 Overall ZAR/USD 5 minute data set (01/07/2002 until 31/01/2003) ........ 231
B.2 Partitioned ZAR/USD 5 minute data set ............................. 244
  B.2.1 First Partition (01/07/2002 until 21/08/2002) ................... 244
B.3 Partitioned ZAR/USD 5 minute data set ............................. 257
  B.3.1 Second Partition (22/08/2002 until 31/01/2003) ............... 257
List of Tables

2.1 Glossary comparing Neural Network and Statistical Terminology .................. 26
3.1 Results From Augmented Dicky-Fuller Test for Daily ZAR/USD Returns .......... 36
3.2 Results From Augmented Dicky-Fuller Test for Daily DEM/USD Returns .......... 43
3.3 Results From Augmented Dicky-Fuller Test for (5 min) ZAR/USD Returns .......... 49
3.4 Results From Augmented Dicky-Fuller Test for (15 min) ZAR/USD Returns ........ 56
3.5 Results From Augmented Dicky-Fuller Test for (30 min) ZAR/USD Returns ........ 63
3.6 Results From Augmented Dicky-Fuller Test for (60 min) ZAR/USD Returns ........ 69
4.1 Results from fitting Stable Distribution to Daily and Intraday ZAR/USD Returns 78
4.2 Results from R/S Analysis of Daily and Intraday ZAR/USD Returns ............ 86
5.1 Data subsets for Training, Validation and Generalisation ......................... 120
5.2 SENN Performance Measures on Initial ZAR/USD Data Set ....................... 131
5.3 SENN Performance Measures on Revised ZAR/USD Data Set ...................... 132
5.4 Comparison of Performance Measures when Number of Agents are varied ....... 133
List of Figures

3-1 Time Series Plot of Daily ZAR/USD Exchange Rate ......................... 33
3-2 Time Series Plot of Daily ZAR/USD Returns ............................. 34
3-3 Histogram and Descriptive Statistics for Daily ZAR/USD Returns Time Series ........................ 37
3-4 Normal Quantile-Quantile Plot of Daily ZAR/USD Returns Time Series .................. 39
3-5 Correlogram for Daily ZAR/USD Returns Time Series ................. 40
3-6 Time Series Plot of Daily DEM/USD Exchange Rate ....................... 41
3-7 Time Series Plot of Daily DEM/USD Returns ................................ 42
3-8 Histogram and Descriptive Statistics for Daily DEM/USD Returns Time Series ..................... 43
3-9 Normal Quantile-Quantile Plot of Daily DEM/USD Returns Time Series .................. 45
3-10 Correlogram for DEM/USD Daily Returns Time Series ................. 46
3-11 Time Series Plot of Intraday (5 minute intervals) ZAR/USD Exchange Rate .................. 47
3-12 Time Series Plot of Intraday (5 minute intervals) ZAR/USD Returns ........................ 48
3-13 Histogram and Descriptive Statistics for Intraday (5 minute intervals) ZAR/USD Returns Time Series .................. 49
3-14 Normal Quantile-Quantile Plot of the Intraday (5 minute interval) ZAR/USD Returns Time Series .................. 51
3-15 Correlogram for ZAR/USD Intraday (5 minute intervals) Returns Time Series .................. 52
3-16 Time Series Plot of Intraday (15 minute intervals) ZAR/USD Exchange Rate .................. 54
3-17 Time Series Plot of Intraday (15 minute intervals) ZAR/USD Returns ........................ 55
3-18 Histogram and Descriptive Statistics for Intraday (15 minute intervals) ZAR/USD Returns Time Series .................. 56
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-19 Normal Quantile-Quantile Plot of Intraday (15 minute interval)</td>
<td>58</td>
</tr>
<tr>
<td>Return Time Series</td>
<td></td>
</tr>
<tr>
<td>3-20 Correlogram for ZAR/USD Intraday (15 minute intervals) Returns</td>
<td>59</td>
</tr>
<tr>
<td>Time Series</td>
<td></td>
</tr>
<tr>
<td>3-21 Time Series Plot of Intraday (30 minute intervals) ZAR/USD</td>
<td>61</td>
</tr>
<tr>
<td>Exchange Rate</td>
<td></td>
</tr>
<tr>
<td>3-22 Time Series Plot of Intraday (30 minute intervals) ZAR/USD</td>
<td>62</td>
</tr>
<tr>
<td>Returns Time Series</td>
<td></td>
</tr>
<tr>
<td>3-23 Histogram and Descriptive Statistics for Intraday (30 minute</td>
<td>63</td>
</tr>
<tr>
<td>intervals) ZAR/USD Returns Time Series</td>
<td></td>
</tr>
<tr>
<td>3-24 Normal Quantile-Quantile Plot of Intraday (30 minute interval)</td>
<td>65</td>
</tr>
<tr>
<td>Returns Time Series</td>
<td></td>
</tr>
<tr>
<td>3-25 Correlogram for ZAR/USD Intraday (30 minute intervals) Returns</td>
<td>66</td>
</tr>
<tr>
<td>Time Series</td>
<td></td>
</tr>
<tr>
<td>3-26 Time Series Plot of Intraday (60 minute intervals) ZAR/USD</td>
<td>68</td>
</tr>
<tr>
<td>Exchange Rate</td>
<td></td>
</tr>
<tr>
<td>3-27 Time Series Plot of Intraday (60 minute intervals) ZAR/USD</td>
<td>69</td>
</tr>
<tr>
<td>Returns Time Series</td>
<td></td>
</tr>
<tr>
<td>3-28 Histogram and Descriptive Statistics for Intraday (60 minute</td>
<td>70</td>
</tr>
<tr>
<td>intervals) ZAR/USD Returns Time Series</td>
<td></td>
</tr>
<tr>
<td>3-29 Normal Quantile-Quantile Plot of Intraday (60 minute interval)</td>
<td>72</td>
</tr>
<tr>
<td>Returns Time Series</td>
<td></td>
</tr>
<tr>
<td>3-30 Correlogram for ZAR/USD Intraday (60 minute intervals) Returns</td>
<td>73</td>
</tr>
<tr>
<td>Time Series</td>
<td></td>
</tr>
<tr>
<td>4-1 Probability plots for daily ZAR/USD returns time series</td>
<td>79</td>
</tr>
<tr>
<td>4-2 Probability plots for intraday (60 min intervals) ZAR/USD returns</td>
<td>80</td>
</tr>
<tr>
<td>time series</td>
<td></td>
</tr>
<tr>
<td>4-3 Probability plots for intraday (30 min intervals) ZAR/USD returns</td>
<td>81</td>
</tr>
<tr>
<td>time series</td>
<td></td>
</tr>
<tr>
<td>4-4 Probability plots for intraday (15 min intervals) ZAR/USD returns</td>
<td>82</td>
</tr>
<tr>
<td>time series</td>
<td></td>
</tr>
<tr>
<td>4-5 Probability plots for intraday (5 min intervals) ZAR/USD returns</td>
<td>82</td>
</tr>
<tr>
<td>time series</td>
<td></td>
</tr>
<tr>
<td>4-6 Observed versus Expected R/S Values for Daily ZAR/USD Returns.</td>
<td>87</td>
</tr>
<tr>
<td>4-7 Observed versus Expected R/S Values for Intraday (60 min) ZAR/USD</td>
<td>88</td>
</tr>
<tr>
<td>Returns.</td>
<td></td>
</tr>
<tr>
<td>4-8 Observed versus Expected R/S Values for Intraday (30 min) ZAR/USD</td>
<td>89</td>
</tr>
<tr>
<td>Returns.</td>
<td></td>
</tr>
<tr>
<td>4-9 Observed versus Expected R/S Values for Intraday (15 min) ZAR/USD</td>
<td>90</td>
</tr>
<tr>
<td>Returns.</td>
<td></td>
</tr>
<tr>
<td>4-10 Observed versus Expected R/S Values for Intraday (5 min) ZAR/USD</td>
<td>91</td>
</tr>
<tr>
<td>Returns.</td>
<td></td>
</tr>
<tr>
<td>5-1 Basic mathematical model of an artificial neuron.</td>
<td>112</td>
</tr>
<tr>
<td>5-2 SENN Model Topology</td>
<td>116</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>5-3</td>
<td>Tanh and Sigmoid Activation Functions</td>
</tr>
<tr>
<td>5-4</td>
<td>SENN Correlation Analysis</td>
</tr>
<tr>
<td>5-5</td>
<td>SENN Sensitivity Analysis</td>
</tr>
<tr>
<td>5-6</td>
<td>SENN Error Progression Monitor</td>
</tr>
<tr>
<td>5-7</td>
<td>SENN Network Evolution Monitor</td>
</tr>
<tr>
<td>5-8</td>
<td>Daily ZAR/USD Exchange Rate Time Series</td>
</tr>
<tr>
<td>6-1</td>
<td>Kalman Filter Estimation Output for DEM/USD returns - ARMA(1,1) model</td>
</tr>
<tr>
<td>6-2</td>
<td>Plotted DEM/USD Actual and Predicted Signals and Residuals</td>
</tr>
<tr>
<td>6-3</td>
<td>Kalman Filter Estimation Output for DEM/USD multi-vector ARMA(1,1) model</td>
</tr>
<tr>
<td>6-4</td>
<td>Kalman Filter Estimation Output for DEM/USD - Time Varying Random Coefficients with RW Errors Model</td>
</tr>
<tr>
<td>6-5</td>
<td>Plotted DEM/USD Actual and Predicted Signals and Residuals</td>
</tr>
<tr>
<td>6-6</td>
<td>Kalman Filter Estimation Output for DEM/USD - VAR(1) Model without Drift</td>
</tr>
<tr>
<td>6-7</td>
<td>Kalman Filter Estimation Output for ZAR/USD returns - ARMA(1,1) model</td>
</tr>
<tr>
<td>6-8</td>
<td>Plotted ZAR/USD Actual and Predicted Signals and Residuals</td>
</tr>
<tr>
<td>6-9</td>
<td>Kalman Filter Estimation Output for ZAR/USD multi-vector ARMA(1,1) model</td>
</tr>
<tr>
<td>6-10</td>
<td>Kalman Filter Estimation Output for ZAR/USD - Time Varying Random Coefficients with RW Errors Model</td>
</tr>
<tr>
<td>6-11</td>
<td>Kalman Filter Estimation Output for ZAR/USD - VAR(1) Model without Drift</td>
</tr>
<tr>
<td>6-12</td>
<td>Plotted ZAR/USD Actual and Predicted Signals and Residuals</td>
</tr>
<tr>
<td>6-13</td>
<td>Kalman Filter Estimation Output for ZAR/USD 5 min returns - ARMA(1,1) model</td>
</tr>
<tr>
<td>6-14</td>
<td>Plotted ZAR/USD(5 min) Actual and Predicted Signals and Residuals</td>
</tr>
<tr>
<td>7-1</td>
<td>Comparison of the Actual versus Predicted Membership State for the overall 5 min ZAR/USD time series</td>
</tr>
<tr>
<td>7-2</td>
<td>Comparison of the Observed versus Predicted Forex Rate for the overall 5 min ZAR/USD time series</td>
</tr>
<tr>
<td>7-3</td>
<td>Comparison of the Observed versus Predicted Forex Rate for the overall 5 min ZAR/USD time series (Modified Kernel)</td>
</tr>
</tbody>
</table>
7-4 Comparison of the Actual versus Predicted Membership State for the ZAR/USD 1st partition time series .................................................. 202
7-5 Comparison of the Observed versus Predicted Forex Rate for the ZAR/USD 1st partition time series .................................................. 203
7-6 Comparison of the Actual versus Predicted Membership State for the ZAR/USD 2nd partition time series .................................................. 206
7-7 Comparison of the Observed versus Predicted Forex Rate for the ZAR/USD 2nd partition time series .................................................. 206
ACKNOWLEDGEMENTS

I would like to thank the following people for their invaluable contributions towards the completion of my dissertation over the past two years.

First and foremost, my supervisor Dr. Renkuan Guo. His patience, wisdom, encouragement and guidance have been paramount to me over the past two years and have contributed immeasurably to the success of this thesis.

Anita Last from Standard Bank South Africa for providing all the time series data without which this dissertation would not have been possible.

Dr. Ralph Grothmann and Dr. Hans Georg Zimmermann for their patience and assistance in helping me set up and understand the neural network models in the SENN environment. Much of the work done on the neural network chapter would not have been possible without their help.

Professor Casparus G. Troskie who was kind enough to share his wealth of experience on Kalman filter modelling. His insights were invaluable in the Kalman filter chapter. His student, Allan Clarke, for the additional Kalman filter references.

Dr. Peter Ouwehand for his efforts in setting up and running the Masters in Financial Mathematics programme at UCT.


Peter Kao, a fellow student, for his willingness to help whenever he could.

Nersan Naidoo, for his invaluable assistance by granting me unlimited access to his colour printer, CD writer, and additional computer.

The Department of Statistical Sciences as well as the Department of Mathematics and Applied Mathematics at UCT for all their support and help.

My manager at work, Mr. Andre Rheeder for his flexibility and support and allowing me time off when it was most needed.

My family for their support and encouragement that pushed me to persevere when I was on the brink of despair.
Chapter 1

INTRODUCTION

The aim of this Masters dissertation is to empirically model and explore foreign exchange rates, in particular the South African Rand/United States Dollar (ZAR/USD) exchange rate returns time series, thereby providing a prediction of the ZAR/USD returns.

1.1 Subject of the Dissertation

This Masters dissertation presents alternative methods to empirically model the high frequency South African Rand/United States Dollar (ZAR/USD) exchange rate returns time series. The techniques employed include multi-agent modelling of foreign exchange markets by neural networks, estimation using the Kalman filter and Fuzzy Markov chain modelling. These methods are applied to foreign exchange data, in particular ZAR/USD data, of a high frequency nature (daily and intraday). The other currency of interest in this dissertation is the German Deutsche Mark/United States Dollar (DEM/USD) foreign exchange rate, as the multi-agent neural network approach was previously applied to model DEM/USD foreign exchange rates.

1.2 Objectives of the Dissertation

The two primary aims of this dissertation are as follows:

1. To empirically explore and model the South African Rand/United States Dollar (ZAR/USD) exchange rate returns time series.
2. To provide predictions of the ZAR/USD exchange rate returns.

In order to accomplish the above-mentioned goals, the contributory research objectives for this dissertation are defined as follows:

- To conduct empirical explorations on foreign exchange (forex) data via the traditional statistical methods, which include time plots, histograms, quantile-quantile plots, calculating descriptive statistics and autocorrelations.

- To fit a stable distribution to the ZAR/USD forex returns data.

- To apply the technique of Re-scaled Range analysis (R/S analysis) on the ZAR/USD forex returns data by calculating Hurst exponents and Fractal dimensions.

- To implement a multi-agent neural network to model the daily ZAR/USD returns time series.

- To utilise the Kalman filter to model the ZAR/USD returns time series.

- To implement fuzzy Markov chains to model very high frequency ZAR/USD forex data.

1.3 Scope and Limitations of the Dissertation

This dissertation is empirical in nature as the title suggests. Therefore, in some instances the findings and results do not always conform to the theory. The findings and results are heavily dependent on the various data sets utilized in the different analysis tools. Lengthy historical data sets, in particular high frequency time series, of the South African market are hard to obtain locally and internationally. Although the foreign exchange market trades 24 hours a day, 7 days a week, the data obtained for this dissertation included data points from Monday to Friday only. South African forex tick data was practically impossible to obtain. Analysis of tick data would have provided an interesting comparison to the daily and intradaily data analysis. A further problem is that the period under study exhibits a bias, which we believe contributes to the findings and results. This bias refers to the extreme volatility experienced by the ZAR/USD exchange rate during the period under study.
1.4 Organisation of the Dissertation

- **Chapter 1** provides a brief overview of the dissertation subject, objectives, methods employed, scope and structure.

- **Chapter 2** provides a critical review of the various modelling techniques utilised, in the empirical modelling of high frequency forex data, in prior theory and published research. The discussion of relevant theory, pertaining to the alternative empirical modelling techniques, is detailed in the respective chapters and sections.

- **Chapter 3** illustrates empirical explorations of the foreign exchange market, of a traditional statistical nature. These traditional statistical methods include various graphical analysis tools (time plots, histograms, quantile-quantile plots and autocorrelation graphs), as well as basic descriptive statistics. This analysis is applied to various foreign exchange data sets. We compare the daily South African Rand/United States Dollar (ZAR/USD) exchange rate to the daily German Deutsche Mark/United States Dollar (DEM/USD) exchange rate. Thereafter we analyse the very high frequency ZAR/USD data, the intradaily data of 5, 15, 30 and 60 minute observations.

- **Chapter 4** continues the empirical modelling of the foreign exchange data by fitting a Stable distribution to the ZAR/USD daily and intradaily data. Furthermore we calculate Hurst exponents and Fractal dimensions via the Re-scaled Range method (R/S analysis).

- **Chapter 5** details the multi-agent neural network approach to model foreign exchange markets. The essential idea in this chapter is to model the South African foreign exchange market through the explicit market paradigm which assumes price shifts in the market are determined by the excess demand or supply in the market. We also discuss a more complex neural network implementation in the form of an error correction neural network.

- **Chapter 6** presents modelling of the foreign exchange markets using the linear Kalman filter estimation technique, followed by a brief discussion of the more complex non-linear Kalman filters.

- **Chapter 7** applies fuzzy Markov chain modelling to the very high frequency data set,
i.e., the ZAR/USD 5 minute observations.

- Chapter 8 provides concluding remarks on the modelling of high frequency South African foreign exchange data, based on the analysis and results obtained in this dissertation.

The bibliography lists the references and quotations cited in this dissertation.

The appendices contain codes for the multi-agent neural network computations, as well as step-by-step computations for the fuzzy Markov chain modelling.

1.5 Softwares used in the Dissertation

An array of different software packages was used to construct and analyse the various high frequency time series models that were investigated. The packages are listed below:

- EVIEWS 3.1
- Stable 3.04 Fortran
- Benoit - Fractal Analysis Software
- SENN
- Microsoft Excel
- Visual Basic for Applications

A brief overview of these packages is now provided.

EVIEWS

EVIEWS is a statistical analysis package used for traditional statistical analysis of time series data. We have used EVIEWS for examining the statistical characteristics of the foreign exchange data. Additionally, EVIEWS also provides the functionality to set up the basic Kalman filter models. Enhanced functionality of the Kalman Filter model results depends on the version of EVIEWS used. However for this dissertation we have utilised EVIEWS 3.1. EVIEWS is a recognised and accepted software package in both academic and financial institutions around the world.
Stable 3.04 Fortran

Stable 3.04 Fortran is a software package developed by John P. Nolan from the Math/Stat Department at the American University, Washington, U.S.A. Stable 3.04 Fortran provides the functionality for additional statistical analysis that is not available through EViews. Specifically we have used Stable 3.04 Fortran to fit a stable distribution to the foreign exchange data. Stable 3.04 Fortran was made available for use in this thesis by, John P. Nolan and has been used in the research work of other post graduate students at the University of Cape Town.

Benoit- Fractal Analysis Software

Benoit is a licensed software package used to analyse time series data by utilising the Re-scaled Range (R/S) analysis to calculate Hurst exponents and Fractal dimensions.

SENN

SENN (Simulated Environment for Neural Networks) is a neural network simulator designed by the Siemens Corporation in Germany. It is a neural network simulator that provides a graphical environment for the construction and evaluation of different neural network models. SENN provides the functionality to set up the different layers of a neural network model, to train the model and to evaluate the performance of the model. SENN is a Linux package that is widely respected internationally for the vast array of neural network functionality that it provides.

Microsoft Excel and Visual Basic for Applications

Microsoft Excel is probably the most widely used spreadsheet application around the world. It provides many basic spreadsheet functions as well as a powerful programming engine in the form of Visual Basic for Applications. We have made use of VBA specifically for cleaning and preprocessing the lengthy historical foreign exchange time series data, as well as other market data. Most of the basic calculations for the Fuzzy Markov Chain modelling were also done using Excel.
Chapter 2

LITERATURE REVIEW

2.1 Overview of the Foreign Exchange (Forex) Market

It is said that money never sleeps. This statement is apparent in financial markets, in particular in the foreign exchange (forex) markets because of its depth, versatility and transparency. The sheer size of forex markets is mind boggling: the daily turnover is about $1.5 trillion.

The foreign exchange market is a market where financial paper with a relatively short maturity is traded. In forex markets, paper denominated in a given currency is traded against paper denominated in another currency. There would not be a forex market if every country in the world used the same currency. Changes in forex rates (i.e., relative prices of different currencies) are caused by both deep structural shifts in the respective economies and a variety of less fundamental factors.

A simple transaction in the forex market is the exchange of two currencies at the spot (fluctuating) rate prevailing at the time of the exchange. Exchanges are not conducted between the interested parties but rather through middlemen, i.e., market makers, resulting in the bid-ask spread, i.e., the buying rate is referred to as the bid rate and the selling rate is referred to as the offer rate or ask rate.

The foreign exchange market is the largest financial market worldwide. It involves participants in different geographical locations, time zones, working hours, time horizons and home currencies. Time horizons vary from long-term investors and central banks to intraday dealers, who close their positions every evening. In this highly complex structure, the market partici-
pants are faced with different constraints and use different strategies to reach their decisions, which differ because of their heterogenous profiles.

When asked for a quotation for a given foreign currency, the forex trader will quote two numbers. The first number indicates the price at which the trader is willing to purchase the foreign currency in terms of the local currency, and the second number indicates the price at which the trader is willing to sell the foreign currency in terms of the local currency.

Foreign currency is like any other traded asset, therefore most analysis performed on asset returns can be tested on foreign exchange, since foreign exchange time series exhibit similar behaviour to stock returns time series.

2.2 Stylized Features of Financial Data

Prior empirical studies on financial time series data indicate that there exists a set of properties, common across many instruments, markets and time periods, classified as stylised facts.

Stylised facts for high frequency data include:

- **Non-Gaussian Distribution of Returns**: The study of kurtosis provides preliminary evidence against the Random Walk Hypothesis for forex rates. Kurtosis is a measure of the "fatness" of a probability distribution's tails. It is measured relative to a normal distribution with the same mean and standard deviation. In comparison to a normal distribution, the corresponding leptokurtic distribution's tails are fatter and platykurtic distribution's tails are thinner. Forex rate changes tend to be leptokurtic (Hsieh, 1998), which implies that dramatic market moves have a greater probability of occurring than is predicted by the normal distribution. The existence of leptokurtic forex rates provides evidence against Random Walk Hypothesis. Fat tails simply mean that the distribution of the returns has tails which are heavier than the tails of a normal distribution.

An alternative way to characterize the high frequency returns distribution is to investigate the tails instead of centre of the distribution. Most types of distributions can be classified into three categories (de Haan 1990):

1. Thin-tailed distributions for which all moments exist;
2. Fat-tailed distributions;

3. Bounded distributions which have no tails.

These categories can be distinguished by one parameter, i.e., the tail index. For distributions of category (1) the tail index equals infinity, tail index greater than zero for category (2) and tail index less than zero for category (3). The empirical estimation of the tail index and the variance of this tail index crucially depend on the sample size. Using too many observations introduces a bias in the tail index as some of the observations do not belong to the tail anymore but are from the centre of the distribution, whilst using too few observations introduces an inefficiency in the estimation of the variance of the tail index. Therefore, the very large sample size available with intraday data ensures that enough tail observations are present in the sample.

- **Volatility Clustering**: The term volatility clustering refers to the phenomenon that large price variations are more likely to be followed by large price variations than smaller ones.

- **Asymmetry-Symmetry**: There is evidence that the distribution of stock market returns are slightly negatively skewed. One could reason that traders react more strongly to negative information than positive information.

- **Negative First-Order Autocorrelation of Returns**: In very high frequency return series, one actually observes negative autocorrelation at lag one, which is traditionally attributed to the bid-ask bounce: transaction prices may occur close to the ask or the bid price and tend to bounce between these two limits.

- **Leverage Effect**: The volatility of financial markets may adapt differently to positive and negative shocks. It is often observed that while markets might remain more or less stable when large positive earnings have been achieved, but when huge losses are experienced, markets become more unpredictable in the periods ahead.

Since we have identified stylised facts of high frequency data, we now analyse previous research on high frequency time series modelling.
2.3 High Frequency Data

The growing availability of tick-by-tick financial data from a variety of liquid markets has prompted trading, risk and asset management to incorporate high frequency data into their systems to improve their performance.

The statistical characteristics of high frequency financial data present many problems to traditional econometric and time series modelling techniques. Standard methodologies cannot cope with erratic data arrival, unusually heavy-tailed distributions and non-synchronicity of information arrival across different markets. A frequently used method to deal with this irregularity is to sample data at random regular intervals of time (i.e. 5-minute, 15-minute, etc.) or derive a homogeneous time series based on some tick time scale (every n ticks for instance).

In this dissertation high frequency data refers to financial time series sampled at intervals smaller than or equal to daily, for e.g. hourly, minute by minute or even tick by tick (i.e. as data reaches the market). There is some discrepancy in the market as well as in research as to whether high frequency data includes daily data or not, but for the purposes of this research we have considered daily data to be high frequency data.

Information arrives continuously in the market and therefore discrete low frequency sampling would ignore a substantial amount of information. The analysis of high frequency sampling aims to reduce the inconsistency between the actual market information flow and its analysis.

2.4 Kalman Filters to Model Financial Time Series

Moody and Wu (1995) examined the correlation structures present in high frequency FX data (USD/DEM) on various time scales and found: negative correlations at lag one in successive tick-by-tick price changes, positive autocorrelations on longer time scales of up to 60 ticks (about 15 min on average for USD/DEM), their proposed measure of uncertainty (tick-by-tick forecast error) correlates with volatility and bid/ask spreads.

Moody and Wu (1995) propose state-space models to analyse high frequency financial time series in which the observed price is a noisy version of an unobserved, less noisy true price process. These stochastic processes include random walks, random or autoregressive trends and
moving average process (MA1).

Mankiw and Miron (1986) estimate the spread between the 3 and 6 month US treasury bill rate. The model is based on the expectations hypothesis for the term structure. This model implies that the longer interest rate would be an average of the current short rate and the expected short rate for the next period plus a term premium. The expected rate plus a forecast error equals the actual rate. Mankiw and Miron specify the term premium as a constant. This may be valid in the short term but their estimation period is about 20 years.

Bolland and Connor (1995) propose a statistical model that uses a state-space representation of time series data in real time. The Kalman filter can easily be enhanced to cope with missing data and data aggregation. This model is not limited to univariate data and can utilise information from various time series.

The methodology presented describes the underlying dynamics of the financial tick data as the states of a state space model (Bolland and Connor 1995). These underlying states (‘true price’) are evolving every second according to their stochastic dynamics, but they are only observed at random intervals via a noisy observation of a quotation or price (somebody’s opinion of the true price). An augmented Kalman filter, using a missing data framework, estimates the model. Since the data is only provided in quantised time steps (i.e. seconds) the state-space model and Kalman filter utilised are discrete. The methodology can be extended to continuous time problems with the Kalman-Bucy filter (Meditch 1969).

There are various causes of observation noise in high frequency financial data. The true price is bounded by a bid and ask price, which are the actual traded prices. Then the observation noise about the true price can be seen as the difference between the true price and the bid and ask prices. Each of the indicative prices observed is a market participant’s opinion of the ‘true price’. It is often the case that several quotes are observed simultaneously resulting from the distribution of opinion within the markets. In practice, prices are rounded to a certain level of accuracy (i.e. 3 or 4 decimal places), which induces noise in observed quotations.

Since the state-space framework is very general, many stochastic models such as random walk, linear, auto-regressive integrated moving average (ARIMA), and non-parametric non-linear models can be embedded within the extended Kalman filter. Bolland and Connor (1995) estimate the state dynamics by a neural network model which does not impose a parametric as-
fractional Brownian motions. Kalman filters are used to estimate random walk and random trend models, and estimation maximization (EM) algorithms are used to estimate the fractional Brownian motion models. To further separate observational noise from the true prices, Moody and Wu use independent component analysis since the observational noise and process noise components of the observed prices are non-Gaussian.

These models were found to be able to explain the observed correlation structures in the USD/DEM exchange rates. Moody and Wu’s statistical modelling of high frequency intraday FX price series demonstrates that trends may in fact exist in noisy FX data and that conventional random walk models of efficient market theory do not explain the correlation structures present in high frequency data (Further studies are required).

Moody and Wu (1996) use a linear state-space representation to model price movements but abandon real time and produce a homogenous time series from ordered data. Also the predictions of future prices cannot be related back to real time and there is no simple equivalent for multivariate time series.

Curt Wells (1996) examined the use of Kalman filters in finance. He identified several uses of the Kalman filter including modelling term structure, forward exchange rate premium, real interest rates, and arbitrage pricing theory. His conclusion is that many more models than those few mentioned should be estimated with time-varying coefficients using the Kalman filter technique.

In many economic models, the public’s expectations of the future have important consequences. These expectations are not observed directly, but if they are formed rationally there are certain implications for the time-series behaviours of observed state-space representation; sample applications include Wall (1980), Burmeister and Wall (1982), Watson (1989) and Imrohoroglu (1993).

Wolff (1987) uses the Kalman filter to model the forward exchange rate premia. The forward exchange rate premia are unobservable. In theory, the forward exchange rate should equal the expected spot rate at a future date plus a risk premium. There is an element of uncertainty in the value of the expected future rate, so Wolff defines the difference between the actual and expected spot rate to be a normally distributed stochastic variable. The unobserved risk premia are modelled either as a first order autoregressive process (AR1) or as a first order
sumption on the relationships, allowing for the dynamics to evolve in nonlinear ways. Classical parametric statistics (frequentist or Bayesian) derive optimal procedures under exact parametric models; however, they give no measure of the behaviour of the models when the assumptions are only approximately valid. With only slight deviations of the assumptions, many procedures break down, thus the estimated models, parameters and tests become meaningless (cf. Tukey 1960).

Robust procedures are those procedures that behave almost optimally when the assumptions are violated. Robustly estimated models and parameters are capable of estimating the structure fitting the bulk of the data as they are not excessively influenced by any minority of the data. Procedures can be made robust to various violations of the assumptions (non-Gaussian residuals, outliers, heteroscedasticity, level shifts, etc.). There are several ways to achieve robustness for heavy tailed residuals: generalised maximum likelihood estimators (M-estimators) assume heavy tailed residual distributions (Huber 1964); R-estimators which use rank of the absolute values to trim heavy tails (Lehman 1975); L-estimators use weighted linear combination of order statistics. Bolland and Connor (1995) focus on robustness of multivariate nonlinear time series models to additive outliers in the observation equation.

Additive outliers in time series are problematic, as each outlier will occur both as inputs to the models and as a target. Filtering outliers from the time series achieves robustness. The methodology presented is based on non-Gaussian filtering and is robust to both input and output space outliers (cf. Masreliez 1975; Kitagawa 1987; Boland and Connor 1995).

Bolland and Connor (1995) also present a neural network extended Kalman filter for modelling noisy financial time series. Neural networks have been successfully applied to the prediction of a diverse range of time series. Neural networks have many preferred features compared to other statistical methods and have been shown to yield significant performance improvements over these statistical methods. No parametric assumption, about the relationship being modelled, is necessary when applying neural nets. Also they are capable of modelling interaction effects, and the smoothness of fit is locally conditioned by the data.

When observations are missing, the neural networks predictions are iterated, which is a discrete time recurrent neural network since the iterated feedforward neural network predictions are based on previous predictions. Bolland and Connor (1995) implement the constrained neural
network to model foreign exchange data, which yields better results since the extended Kalman filter is guaranteed to have bounded errors if the neural network model is 'observable and controllable' which is enabled by the existence of a stable fixed point at the origin. Thus it reflects their belief that price increments beyond a certain horizon are unpredictable for the Dollar-Deutsche Mark exchange rate. In the presence of missing data, as is the case with foreign exchange data, the performance of the constrained neural network is shown to be quantitatively and qualitatively superior to the unconstrained neural network.

2.5 Neural Networks to Model Financial Time Series

The following glossary has been extracted from Trigueiros & Taffler (1996, pg 355). Its explains neural network terminology through the equivalent statistical term.

<table>
<thead>
<tr>
<th>Neural Network terminology</th>
<th>Statistical modelling terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Network</td>
<td>Model</td>
</tr>
<tr>
<td>Synapses, weights, connectivities, etc.</td>
<td>Coefficients of the model</td>
</tr>
<tr>
<td>Inputs</td>
<td>Explanatory variables</td>
</tr>
<tr>
<td>Outputs</td>
<td>Response variables</td>
</tr>
<tr>
<td>Outcome or target</td>
<td>Expected value</td>
</tr>
<tr>
<td>Node</td>
<td>Logistic regression</td>
</tr>
<tr>
<td>Hidden layer</td>
<td>Intermediate set of logistic regressions</td>
</tr>
<tr>
<td>Learning</td>
<td>Coefficient estimation</td>
</tr>
<tr>
<td>Supervised learning</td>
<td>Regression, discriminant analysis, etc.</td>
</tr>
<tr>
<td>Unsupervised learning</td>
<td>Principal components and cluster analysis</td>
</tr>
<tr>
<td>Architecture</td>
<td>Model description</td>
</tr>
<tr>
<td>Convergence</td>
<td>In-sample performance</td>
</tr>
<tr>
<td>Generalisation</td>
<td>Out-of-sample performance</td>
</tr>
</tbody>
</table>

Table 2.1: Glossary comparing Neural Network and Statistical Terminology

2.5.1 Multi-Agent Modelling of FX-Markets by Neural Networks

Several publications have attempted to model financial markets on the basis of multi-agent theory. A market can be perceived as a network where a large number of agents interact with one another. These agents evaluate sets of external information, e.g. fundamental or technical indicators, in an attempt to maximize their profit by anticipating prospective changes in the market prices. Market equilibrium conditions and market price formation can be depicted on
the basis of agents trading activities, i.e. agents buying/selling decisions which can be summed up to the aggregated demand/supply of the market.

A common trait of these multi-agent approaches is that markets are simulated under laboratory conditions. The authors experiment when studying the behaviour or learning of the agents, and different trading strategies. As a result, these models can only produce artificial time series as outputs, with characteristics of real world price dynamics, e.g. effects of volatility clustering. Most of these models are unable to forecast real world price movements of financial markets.

Lux and Marchesi (1999) present a stochastic simulation model of an artificial financial market where agents interact. These agents are divided into fundamentalists and noise traders. The fundamentalists follow the Efficient Market Hypothesis, i.e. they expect the price to follow the fundamental value of the asset (discounted sum of expected future earnings), whilst the noise traders attempt to identify price trends and patterns (charts). Since it is important for the resulting market, noise traders are further divided into optimists (believe in a rising market) and pessimists (believe in a declining market). The dynamics of the artificial market comprise the following elements: agents changing their views from one group to another, e.g. pessimistic to optimistic or noise traders to fundamentalists; fundamental value changes; and endogenous price changes. Price adjustments are responses of the market to imbalances between demand and supply, which originate from the agents’ decisions. This multi-agent model allows for an artificial time series exhibiting real world empirical data, i.e. extreme events and volatility clustering.

Beltratti and Margarita (1992) and Beltratti, Margarita and Terna (1996) use neural networks to model price forecasts and agents forecasting ability. The neural networks estimate market price forecasts using external information, lagged and averaged historical prices. Trade occurs in a random matching environment. Agents are randomly matched and trade occurs when pairs have opposing expectations of the market price, i.e. one agent anticipates a raising market price while the other expects a declining market price. They then trade at the price in between their two expected values. When modelling the agents’ behaviour the complexity of the underlying neural networks are varied by changing the number of hidden units in the networks. The neural nets are divided into ‘smart’ and ‘naive’, i.e., the intelligent traders (smart) were

27
assigned more hidden units at a greater cost than their naive counterparts.

Zimmermann, Neuneier and Grothmann (1997) use a multi-agent approach, which allows for the fitting of real world behaviour, to model the FX market. Econometrical and quantitative techniques provided by the fields of mathematics and statistics allow for this fitting of real world behaviour. Because of their ability to approximate any functional structure, neural networks allow the fitting of high dimensional nonlinear models (Hornik, Stinchcombe and White, 1989). Neural networks represent a framework for modelling complex economical systems as they allow the construction of models able to handle high dimensional and nonlinear interactions simultaneously.

The authors link neural networks and economic theory in the following way: A single neuron can be interpreted as an elementary economic decision making process, because it unites the process of information filtering, the formation of a superposition of information and the execution of an action. From this point of view, a neural network, which may include hundreds of neurons, reflects the interaction of a large number of decision makers representing a market process (Neuneier and Zimmermann, 1998).

Zimmermann, Neuneier and Grothmann (1997) attempt to establish a link between econometric and economic (artificial) market modelling by the application of neural networks, which integrate the micro-economic decision behaviour of single agents into one macro-economic market model. Agents in a single market act independently from each other (micro-structure). If their trading activities match exactly, the market is in an equilibrium state. Agents decisions are aggregated on the macro-level, and if aggregated demand is unequal to aggregated supply the market is in a disequilibrium state. Adjusting the price level will correct this market disequilibrium. If there is an excess demand a price increase will bring the market back into an equilibrium state (Agents prefer to sell goods as a higher price, while the purchasing agents will withdraw from the market). Similarly lowering the market price re-balances an excess supply (Agents planning to sell will withdraw from the market and the buying agents will increase).

Zimmermann, Neuneier and Grothmann (1997) apply this market mechanism to forecast FX-rates, in particular, on the DEM / USD FX-market (US dollars priced in German Mark). Agents try to maximize their profits by anticipating changes in the FX-rate, for example in the DEM / USD FX-market, agents believing in a devaluation of DEM would invest in USD, while
agents anticipating a devaluation of the USD would invest in DEM. The agent can therefore make one of three decisions: buy DEM / sell USD, sell DEM / buy USD, neither buy nor sell. In order to make a decision the agent will have to filter information relevant to the FX-rate, thereafter analyse that information, finally executing a decision based on his prediction of the market.

In summary, there is a fair amount of research on empirical modelling of high frequency data as well as forex data. However, it is more common to find research on low frequency data sets rather than very high frequency time series for the reasons discussed previously. There is perhaps lesser research on the modelling of the South African forex market, especially that of high frequency data.
Chapter 3

EXPLORATIONS OF FOREX DATA USING TRADITIONAL STATISTICAL METHODS

The aim of this chapter is to conduct empirical explorations on foreign exchange (FX) data using traditional statistical methods.

Before we proceed to the statistical explorations of the forex time series, we briefly define a few time series concepts.

3.1 Concepts of Statistical Time Series

Random Variables

Let \((\Omega, \mathcal{M}, P)\) be a probability space, where \(\Omega\) is the sample space (set of all elementary events), \(\mathcal{M}\) is a sigma-algebra of events or subsets of \(\Omega\), and \(P\) is a probability measure defined on \(\mathcal{M}\).

Definition 1 A random variable is a real valued function \(y : \Omega \rightarrow \mathbb{R}\) such that for each real number \(c\),

\[
A_c = \{ \omega \in \Omega \mid y(\omega) \leq c \} \in \mathcal{M}.
\]
Stochastic Processes

Suppose $\mathcal{T}$ is some index set with at most countably many elements, for example, the set of all integers or all positive integers.

**Definition 2** A (discrete) stochastic process is a real valued function

$$y : \mathcal{T} \times \Omega \to \mathbb{R}$$

such that for each fixed $t \in \mathcal{T}$, $y_t(\omega)$ is a random variable.

A stochastic process is an ordered sequence of random variables $\{y_t(\omega), \omega \in \Omega, t \in \mathcal{T}\}$, such that for each $t \in \mathcal{T}$, $y_t(\omega)$ is a random variable on the sample space $\Omega$, and that for each $\omega \in \Omega$, $y_t(\omega)$ is a realization of the stochastic process on the index set $\mathcal{T}$ (that is an ordered set of values, each corresponding to a value of the index set).

**Definition 3** A time series $\{y_t\}_{t=1}^T$ is (the finite part of) a particular realization $\{y_t\}_{t \in \mathcal{T}}$ of a stochastic process.

A realization of a stochastic process is a function $\mathcal{T} \to \mathbb{R}$ where $t \to y_t(\omega)$. The underlying stochastic process is said to have generated the time series. The time series $y_1(\omega), \ldots, y_T(\omega)$ is usually denoted by $y_1, \ldots, y_T$ or simply by $\{y_t\}$.

**Stationarity**

**Definition 4** The process $\{y_t\}$ is said to be weakly stationary of order 2 if the first and second moments of the process exist and are time-invariant:

$$E[y_t] = \mu < \infty \quad \text{for all } t \in \mathcal{T}$$

$$E[(y_t - \mu)(y_{t-h} - \mu)] = \gamma(h) < \infty \quad \text{for all } t \text{ and } h$$

Stationarity implies $\gamma_t(h) = \gamma_t(-h) = \gamma(h)$.

**Definition 5** The process $\{y_t\}$ is said to be strictly stationary if for any values of $h_1, h_2, \ldots, h_n$ the joint distribution of $(y_t, y_{t+h_1}, \ldots, y_{t+h_n})$ depends only on the intervals $h_1, h_2, \ldots, h_n$ but
not on the time \( t \) itself:

\[
f(y_t, y_{t+h_1}, \ldots, y_{t+h_n}) = f(y_{\tau}, y_{\tau+h_1}, \ldots, y_{\tau+h_n}) \quad \text{for all } t \text{ and } \tau.
\]

Strict stationarity implies that all existing moments are time-invariant.

**Definition 6** The process \( \{y_t\} \) is said to be **Gaussian** if the joint density of \( (y_t, y_{t+h_1}, \ldots, y_{t+h_n}) \)

\[
f(y_t, y_{t+h_1}, \ldots, y_{t+h_n})
\]

is Gaussian for any \( h_1, h_2, \ldots, h_n \).

### 3.2 Data Set

The FX data utilised in this chapter was obtained from Standard Bank, kindly provided by Anita Last. The data set comprises the South African Rand/United States Dollar (ZAR/USD) and German Deutsche Mark/United States Dollar (DEM/USD) FX rates. The convention for the denotation of the FX rate for this dissertation is illustrated by the following example, i.e. if the ZAR/USD rate is 8.00 then 1 US Dollar(USD) is equivalent to 8 SA Rands(ZAR).

Although the FX market trades 24 hours a day, seven days a week, the available FX data sets only comprised data points from Monday to Friday. The ZAR/USD data sets include a daily observation data set, as well as a 5 minute observation data set. One or more sets of 15 minute, 30 minute and 60 minute observations were extracted from the 5 minute data set. The DEM/USD data series consists of daily data. The daily observations are recorded as the last traded FX rate observed for the respective dates. The 5 minute observations are the intraday FX rates recorded after every five minutes.

The ZAR/USD daily data sample is taken from 02/01/1990 until 28/02/2003, whilst the ZAR/USD intraday data sample (5, 15, 30 and 60 minute observations) is from 01/07/02 until 31/01/2003. The DEM/USD daily data sample is taken over a shorter period, i.e. from 02/01/1990 until 31/12/1998, as a consequence of the Deutsche Mark being replaced by the Euro, with the phasing out of the Deutsche Mark occurring from 1999.
3.3 Statistical Analysis of Daily ZAR/USD Time Series

3.3.1 Time Series Plots

Figure 3-1 below shows the time plot of the daily ZAR/USD exchange rate time series generated by EViews:

![Time Series Plot of Daily ZAR/USD Exchange Rate](image)

From 3-1 above, the following observations can be made:

1. In the long term, the Rand has moved in one general movement in favour of weakness against the US Dollar.

2. There is an exponentially increasing trend resulting in a sharp spike (weakening of the Rand), thereafter there is a decline in the trend (strengthening of the Rand) due to the overcorrection of the rand. The daily ZAR/USD time series is not weakly stationary.

3. During the last few years the Rand/Dollar exchange rate has become very volatile.

The daily exchange rate is then transformed into daily log returns, i.e. the (continuously
compounded) log return on the daily exchange rate over the time interval \([t-1, t]\) is defined as:

\[ r_t = p_t - p_{t-1} = \ln P_t - \ln P_{t-1} \]

where \(P_t\) is the exchange rate at time period \(t\).

In this dissertation, we shall refer to the log returns time series as the returns time series.

Figure 3-2 below shows the time plot of the daily ZAR/USD exchange rate returns time series generated by EViews:

![Time Series Plot of Daily ZAR/USD Returns](image)

**Figure 3-2: Time Series Plot of Daily ZAR/USD Returns**

From 3-2 above, the following observations can be made:

1. The daily returns time series looks relatively stationary.

2. Relatively large positive \(r_t\) values occur when \(p_t\) is substantially greater than \(p_{t-1}\), which implies an increase in the daily exchange rate from \(t-1\) to \(t\) which implies a weakening of the Rand against the Dollar.

3. Relatively large negative \(r_t\) values occur when \(p_t\) is substantially less than \(p_{t-1}\), which implies a decrease in the daily exchange rate from \(t-1\) to \(t\) which implies a strengthening of the Rand against the Dollar.
4. The extreme (technical) positive returns are slightly larger in magnitude than the extreme (technical) negative returns, where extreme positive returns reflect depreciating returns of the Rand and extreme negative returns reflect appreciating returns of the Rand. During the period under study it was evident that the Rand experienced greater depreciation extremes than appreciation extremes.

5. Most of these extreme (technical) returns occur during the latter half of the time series. Prior to the 1700th trading day extreme returns were rare. Over this period, extreme returns occurred between February 1996 and April 1996 (trading day interval 1574 to 1623) when the Rand lost approximately 20% of its value against the US Dollar.

6. Around September/October 1997, the world witnessed the start of the so-called Asian crisis. This crisis affected all emerging markets in May 1998 and the Rand was materially affected (trading day interval 2187 to 2232).

7. From the beginning of 2000 to 11 September 2001 (trading day interval 2585 to 3025) the Rand maintained an almost consistent and fairly well-defined declining trend against the US Dollar.

8. Thereafter (trading day interval 3025 to 3403) the Rand depreciation reached its trough with a new all-time low on 21 December 2001. The political instability in Zimbabwe, the AIDS crisis, and the political shock after September 11th attacks on the US contributed to the depreciation of the Rand. Brazil’s default on payment of their loan to the International Monetary Fund (IMF) caused investor confidence in emerging markets to falter. After the sharp depreciation of the Rand, the Rand started to stabilise as can be seen from the extreme negative returns (appreciation) during this time period.

3.3.2 Test for Stationarity

The unit root test is conducted in EViews to ensure that the daily returns time series is indeed stationary.

Table 3.1 below contains the results of the Augmented Dickey - Fuller (ADF) test on the daily ZAR/USD exchange rate returns time series:
<table>
<thead>
<tr>
<th>ADF Test Statistic</th>
<th>-26.62197</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacKinnon critical values for rejection of $H_0$:</td>
<td>1% Critical Value = -3.4354</td>
</tr>
<tr>
<td></td>
<td>5% Critical Value = -2.8629</td>
</tr>
<tr>
<td></td>
<td>10% Critical Value = -2.5675</td>
</tr>
</tbody>
</table>

Table 3.1: Results From Augmented Dicky-Fuller Test for Daily ZAR/USD Returns

From Table 3.1 above, it is noted that:

$$ADF \text{ Test Statistic} = -26.62197 < -3.4354 \text{ (Critical Value at 1%).}$$

Thus, the null hypothesis of a unit root (or non-stationary) time series is rejected far beyond the 1% level, which implies that there is strong evidence to suggest that the observed daily returns time series is stationary.

### 3.3.3 Histogram and Descriptive Statistics

The histogram graphically shows the following:

1. centre (i.e., the location) of the data;
2. spread (i.e., the scale) of the data;
3. skewness of the data;
4. presence of outliers; and
5. presence of multiple modes in the data.

Figure 3-3 below shows the histogram for the daily ZAR/USD exchange rate returns times series generated by EVIEWS, and some of the important descriptive statistics of the data.
Figure 3-3: Histogram and Descriptive Statistics for Daily ZAR/USD Returns Time Series

From Figure 3-3 above, the following observations can be made:

1. The daily returns series has a small positive mean daily return of 0.0060337 (0.034%), which is in agreement with the fact that the Rand depreciated during the time period investigated.

2. The median daily return of 0.000149 (0.015%) is less than the mean daily return, meaning that there are more positive daily returns. This distribution shows a longer right tail than left tail.

3. The standard deviation of the returns is 0.006879 (0.5879%), which implies that the largest negative return (appreciation of the Rand) is 7.338 standard deviations away from the mean daily return and the largest positive return (depreciation of the Rand) is 8.917 standard deviations away from its mean. One would therefore expect the returns series not to be normally distributed.

4. The range of the observed daily returns time series is 0.112488 (11.249%), since

\[ \text{Range} \left( \{ r_i \} \right) = \max \{ r_i \} - \min \{ r_i \} = 0.061675 - (-0.050813) \]

Thus, the daily returns have a wide range of values. The standard deviation of the daily returns is relatively small in comparison to the range, resulting from a large number of daily returns close in value to the mean daily return thereby offsetting the extreme
positive and negative daily returns, suggesting that the distribution of daily returns time series is sharply peaked at the mean with heavy-tails.

5. The coefficient of skewness of 0.717783 > 0 indicates that the daily returns time series distribution is asymmetric about its mean, and is more positively skewed than the corresponding normal distribution (which has a coefficient of skewness of 0). This distribution has tails that extend further towards the right than to the left due to the occurrence of more extreme positive returns than extreme negative returns.

6. The coefficient of kurtosis of 16.90105 > 3 indicates that the distribution is significantly leptokurtic (i.e. sharp peaked and heavy-tailed as compared to the corresponding normal distribution, which has a coefficient of kurtosis of 3).

7. The Jarque-Bera test statistic value measures the difference of the skewness and kurtosis of the time series with those from the normal distribution. At 27391.93 it is far larger than the critical chi-squared value ($\chi^2_{0.05,2}$) of 5.99, which is confirmed by an observed p-value of 0.000000, implying that null hypothesis of normality is rejected at the 5% (and even far beyond the 1%) significance level. Thus, there is very strong evidence to suggest that the distribution of the daily returns time series is non-Gaussian.

### 3.3.4 Normal Quantile-Quantile Plots

The normal quantile-quantile (Q-Q) plot is a graphical technique for assessing whether or not a data set is approximately normally distributed. It plots the empirical quantiles of a series against the quantiles of a corresponding normal distribution. If the two distributions are the same, the fitted Q-Q plot should lie on a straight 45° line. Departures from this straight line indicate departures from normality. Figure 3-4 below shows the normal Q-Q plot for the observed daily returns time series computed by EViews.
Figure 3-4: Normal Quantile-Quantile Plot of Daily ZAR/USD Returns Time Series

From Figure 3-4 above, the following observations can be made:

1. The distribution of the ZAR/USD daily returns time series is evidently not normal, as the plot follows a non-linear pattern.

2. The extreme positive quantiles appear to have a higher frequency and magnitude than the extreme negative quantiles, which indicates that the distribution for the daily returns will have a longer right tail than the left tail. We would therefore expect the coefficient of skew for the observed daily returns time series to be positive, since the coefficient of skew is sensitive to outlying observations.

3.3.5 Autocorrelation and Partial Correlation

Figure 3-5 below illustrates the sample autocorrelation function (acf) and partial autocorrelation function (pacf) of the observed daily returns time series generated by EViews for lags up to 15 intervals:
From Figure 3-5 above, the following interpretations can be made:

1. No evidence of exponential decay exists in the acf as lags increase, implying that the observed daily returns time series cannot be explained by a pure autoregressive process of order $k$. The relatively large autocorrelation at higher lags suggests that long memory in the time series may exist.

2. The observed daily returns time series does not have a significant first order acf and pacf.

3. The approximate 95% standard error for both the acf and pacf estimates is $\frac{2}{\sqrt{T}}$, where $T$ is the number of observations in the observed time series. The observed daily returns time series has a sample size of 3403 observations, which is fairly large. As a result, the standard error bound of the estimates will be extremely small ($\pm \frac{2}{\sqrt{3403}} = \pm 0.034284$). Thus, if an acf or pacf estimate of order $k$ has a value exceeding this error bound, it cannot be used as conclusive evidence for significance at order $k$.

4. The reported probabilities (p-values) for the Q-statistic are 0.000 for all lags after lag 6.
The null hypothesis of no autocorrelation up to order $k$ is rejected at all $k = 6, 7, \cdots, 3403$, and the corresponding acf is significant at the 5% level. The $Q$-statistic is dependent on the sample size, and the larger the sample size the larger the value of the $Q$-statistic, which explains why the reported $p$-values after lag 6 are zero. Thus, a significant $Q$-statistic cannot be taken as conclusive evidence.

5. When there is a large sample size, and especially when the sample consists of high frequency data, then classical time series statistical tests can become meaningless and no valid conclusions may be drawn from them.

3.4 Statistical Analysis of Daily DEM/USD Time Series

3.4.1 Time Series Plots

Figure 3-6 below shows the graph of the daily DEM/USD exchange rate time series generated by EViews:

![Figure 3-6: Time Series Plot of Daily DEM/USD Exchange Rate](image)

From Figure 3-6 above, we observe the following:
1. The Deutsche Mark has both periods of weakness and strength against the US Dollar. The currency has been relatively stable over the 8 year period under study. In comparison to the ZAR/USD exchange rate, the DEM/USD exchange rate is much less volatile.

Figure 3.7 below illustrates the time plot of the daily DEM/USD exchange rate returns time series generated by EViews:

![Time Series Plot of Daily DEM/USD Returns](image)

**Figure 3.7: Time Series Plot of Daily DEM/USD Returns**

From figure 3.7 above, the following can be noted:

1. The daily returns time series appears relatively stationary.
2. The magnitude of the extreme DEM/USD returns is much smaller than that of the extreme ZAR/USD returns.
3. This time plot illustrates that the DEM/USD exchange rate is much more stable than the ZAR/USD exchange rate.

### 3.4.2 Test for Stationarity

To test stationarity of the daily returns time series, the unit root test is conducted in EViews.
Table 3.2 below contains the results of the Augmented Dicky-Fuller (ADF) test on the daily ZAR/USD exchange rate returns time series:

<table>
<thead>
<tr>
<th>ADF Test Statistic</th>
<th>-21.61785</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacKinnon critical values for rejection of $H_0$ :</td>
<td></td>
</tr>
<tr>
<td>1% Critical Value</td>
<td>-3.4362</td>
</tr>
<tr>
<td>5% Critical Value</td>
<td>-2.8663</td>
</tr>
<tr>
<td>10% Critical Value</td>
<td>-2.5677</td>
</tr>
</tbody>
</table>

Table 3.2: Results From Augmented Dicky-Fuller Test for Daily DEM/USD Returns

From Table 3.2 above, it is noted that:

$$ADF \text{ Test Statistic} = -21.61785 \ll -3.4362 \text{ (Critical Value at 1%).}$$

Thus, the null hypothesis of a unit root (or non-stationary) time series is rejected at the 1% level. Thus there is strong evidence to suggest that the observed daily returns time series is stationary.

3.4.3 Histogram and Descriptive Statistics

Figure 3-8 below shows the histogram and descriptive statistics for the daily DEM/USD returns time series generated by EViews:

![Histogram and Descriptive Statistics](image)

Figure 3-8: Histogram and Descriptive Statistics for Daily DEM/USD Returns Time Series

From figure 3-8 above, the following can be noted:
1. The DEM/USD histogram looks a lot more symmetrical than the ZAR/USD histogram. It has shorter tails and is less sharply peaked.

2. Given the maximum and minimum of the DEM/USD returns histogram (0.0316720 and -0.030007), the range of 0.061679(6.168%) is approximately half the magnitude of the range of the ZAR/USD returns histogram.

3. The coefficient of skewness of 0.031630 > 0 indicates that the daily returns time series distribution is slightly asymmetric about its mean, and is more positively skewed than the corresponding normal distribution. The USD/ZAR daily returns time series has a larger coefficient of skewness which is expected since the right tail is longer than the left tail of its distribution.

4. The coefficient of kurtosis of 5.154842 > 3 indicates that the distribution is slightly leptokurtic, but not as significantly leptokurtic as the ZAR/USD daily returns distribution.

5. The Jarque-Bera test statistic of 453.8082 is much smaller than that of the ZAR/USD daily returns. Although the distribution of the DEM/USD daily returns time series is non-Gaussian, it does appear to be closer to the normal distribution than the ZAR/USD daily returns distribution.

3.4.4 Normal Quantile-Quantile Plots

Figure 3-9 below shows the normal Q-Q plot for the observed DEM/USD daily returns time series computed by EViews.
Figure 3-9: Normal Quantile-Quantile Plot of Daily DEM/USD Returns Time Series

1. The Q-Q plot appears to be slightly non-linear therefore the distribution of the DEM/USD daily returns time series is not normal.

2. The Q-Q plot of the ZAR/USD daily returns time series is far more non-linear than the Q-Q plot of the DEM/USD daily returns time series.

3.4.5 Autocorrelation and Partial Autocorrelation

Figure 3-10 below illustrates the acf and pacf of the DEM/USD daily returns time series generated by EViews for lags up to 15 intervals.
From figure 3-10 above, the following is noted:

1. The observed daily returns time series does not have a significant first order acf and pacf.

2. No evidence of exponential decay exists in the acf as lags increase, which implies that the observed daily returns time series cannot be explained by a pure autoregressive process of order $k$. The relatively large pacf values at higher lags suggest that long memory in the time series may exist.

We now proceed to the intraday data analysis.

3.5 Statistical Analysis of 5 Minute ZAR/USD Time Series

3.5.1 Time Series Plots

Figure 3-11 below shows the time plot of the intraday (5 minute intervals) ZAR/USD exchange rate time series generated by EViews:
From Figure 3-11 above, the following observations are made:

1. The observed intraday (5 minute intervals) ZAR/USD time series is not (weakly) stationary.

2. The general trend of this volatile time series is that of a decreasing time series, i.e., a strengthening of the Rand.

Figure 3-12 below shows the time plot of the intraday (5 minute intervals) ZAR/USD exchange rate returns time series generated by EViews:
Figure 3.12. Time Series Plot of Intraday (5 minute intervals) ZAR/USD Returns

From Figure 3.12 above, the following characteristics are noted:

1. The observed intraday (5 minute intervals) returns time series looks relatively stationary.

2. It is almost meaningless to comment on the extreme returns occurring in this time series, since the intervals in this very high frequency time series are that of five minutes. The extreme returns observed will also be smaller in magnitude than those of the daily returns series, into which they aggregate.

3.5.2 Test for Stationarity

The unit root test is conducted in EViews to ensure stationarity of the (5 minute interval) ZAR/USD returns times series.

Table 3.3 contains the results of the Augmented Dicky - Fuller (ADF) test on the intraday (5 minute interval) ZAR/USD exchange rate returns time series:

From Table 3.3, it is noted that:

$$ADF \text{ Test Statistic} = -102.9422 < -3.4337 \text{ (Critical Value at 1%).}$$
ADF Test Statistics  
-102.9422

<table>
<thead>
<tr>
<th>MacKinnon critical values for rejection of $H_0$:</th>
<th>1% Critical Value = -3.4337</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5% Critical Value = -2.8622</td>
</tr>
<tr>
<td></td>
<td>10% Critical Value = -2.5671</td>
</tr>
</tbody>
</table>

Table 3.3: Results From Augmented Dickey-Fuller Test for (5 min) ZAR/USD Returns

Thus, the null hypothesis of a unit root (or non-stationary) time series is rejected beyond the 1% level, which implies that there is strong evidence to suggest that the observed returns time series is stationary.

3.5.3 Histogram and Descriptive Statistics

Figure 3-13 below shows the histogram for the intraday (5 minute interval) ZAR/USD exchange rate returns time series generated by EViews, and some of the important descriptive statistics of the data:

![Histogram and Descriptive Statistics](image)

Figure 3-13: Histogram and Descriptive Statistics for Intraday (5 minute intervals) ZAR/USD Returns Time Series

From Figure 3-13 above, the following observations can be made:

1. The returns series has a mean return of -0.00000435, i.e., extremely close to zero.

2. The median return of 0.000000 is slightly greater than the mean return. This implies that there are more positive returns than negative returns.
3. The standard deviation of the 5 minute returns is 0.000805. The largest negative return (appreciation of the Rand) is 16.385 standard deviations away from the mean daily return and the largest positive return (depreciation of the Rand) is 17.091 standard deviations away from its mean, which leads one to suspect that the intraday time series is not normally distributed.

4. The range of the observed 5 minute returns time series is:

\[
\text{Range} \left( \{ r_t \} \right) = \max \{ r_t \} - \min \{ r_t \} = 0.014779 - (-0.014169) = 0.028948 \ (2.895\%)
\]

The range indicates that the returns have a relatively small range of values. The standard deviation of the returns is relatively small in comparison to the range, resulting from a large number of returns close in value to the mean return thereby offsetting the extreme positive and negative returns, suggesting that the distribution of the returns time series is highly-peaked at the mean.

5. The coefficient of skewness of 0.085021 > 0 indicates that the 5 minute returns time series distribution is asymmetric about its mean, and is more positively skewed than the corresponding normal distribution (which has a coefficient of skewness of 0), which implies that the distribution has tails that extend further towards the right than to the left due to the occurrence of more extreme positive returns than extreme negative returns.

6. The coefficient of kurtosis of 24.64544 > 3 indicates that the distribution is significantly leptokurtic (i.e. sharp peaked and heavy-tailed as compared to the corresponding normal distribution, which has a coefficient of kurtosis of 3).

7. The Jarque-Bera test statistic value measures the difference between the skewness and kurtosis of the time series and those of the normal distribution. At \(80246.4\) it is far larger than the critical chi-squared value \(\chi^2_{0.05,2}\) of 5.99, which is confirmed by an observed p-value of 0.000000. This implies that null hypothesis of normality is rejected at the 5% (and even far beyond the 1%) significance level. Thus, there is very strong evidence to suggest that the distribution of the daily returns time series is non-Gaussian.
3.5.4 Normal Quantile-Quantile Plots

Figure 3-14 below shows the normal Q-Q plot for the observed intraday (5 minute intervals) returns time series computed by EViews:

![Normal Quantile-Quantile Plot](image)

From Figure 3-14 above, the following observations can be made:

1. The distribution of the returns time series is evidently not normal, as the plot follows a non-linear pattern.

2. The extreme positive quantiles appear to have a slightly higher magnitude than the extreme negative quantiles, which indicates that the distribution for the returns will have a slightly longer right tail than the left tail. We would therefore expect the coefficient of skew for the observed daily returns time series to be positive, since the coefficient of skew is sensitive to outlying observations.

3. The Q-Q plot is not continuous at 0 since the return series contains very many zeroes.
3.5.5 Autocorrelation and Partial Correlation

Figure 3-15 below illustrates the sample autocorrelation function (acf) and partial autocorrelation function (pacf) of the observed returns time series generated by EViews for lags up to 15 intervals:

<table>
<thead>
<tr>
<th>Sample 1 44063</th>
<th>Included observations: 44063</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autocorrelation</td>
<td>Partial Correlation</td>
</tr>
<tr>
<td>1</td>
<td>-0.227</td>
</tr>
<tr>
<td>2</td>
<td>-0.015</td>
</tr>
<tr>
<td>3</td>
<td>-0.012</td>
</tr>
<tr>
<td>4</td>
<td>0.010</td>
</tr>
<tr>
<td>5</td>
<td>-0.257</td>
</tr>
<tr>
<td>6</td>
<td>0.001</td>
</tr>
<tr>
<td>7</td>
<td>0.002</td>
</tr>
<tr>
<td>8</td>
<td>0.006</td>
</tr>
<tr>
<td>9</td>
<td>0.002</td>
</tr>
<tr>
<td>10</td>
<td>0.002</td>
</tr>
<tr>
<td>11</td>
<td>0.008</td>
</tr>
<tr>
<td>12</td>
<td>0.006</td>
</tr>
<tr>
<td>13</td>
<td>0.000</td>
</tr>
<tr>
<td>14</td>
<td>-0.014</td>
</tr>
<tr>
<td>15</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Figure 3-15: Correlogram for ZAR/USD Intraday (5 minute intervals) Returns Time Series

From Figure 3-15 above, the following interpretations can be made:

1. The lag 1 acf and pacf estimate of -0.227 is far larger than the 95% standard error bounds (as indicated by the dotted line), which implies that the lag 1 acf and pacf estimate is significantly different from zero at the 5% significance level. An acf and pacf value of -0.227 implies that a lag 1 effect only explains 5.15%\((-0.227)^2 = 0.051529\) or 5.15% of the variation.

2. No evidence of exponential decay exists in the acf as lags increase, implying that the observed returns time series cannot be explained by a pure autoregressive process of order k.

52
3. The approximate 95% standard error for both the acf and pacf estimates is \( \frac{2}{\sqrt{T}} \), where \( T \) is the number of observations in the observed time series. The observed returns time series has a sample size of 44063 observations, which is extremely large. As a result, the standard error bound of the estimates will be extremely small \( (\pm \frac{2}{\sqrt{44063}} = \pm 0.009528) \). Thus, if an acf or pacf estimate of order \( k \) has a value exceeding this error bound, it cannot be used as conclusive evidence for significance at order \( k \).

4. At lag 1 the Q-statistic has an extremely large value of 2260.8. The reported probabilities (p-values) for the Q-statistic are 0.000 for all lags. Therefore the null hypothesis of no autocorrelation up to order \( k \) is rejected at all \( k = 1, 2, \ldots, 44063 \), and the corresponding acf is significant at the 5% level. The Q-statistic is dependent on the sample size, and the larger the sample size the larger the value of the Q-statistic, which explains why all the reported p-values are zero. Thus, a significant Q-statistic cannot be taken as conclusive evidence for a significant order \( k \) acf.

5. When there is a large sample size, and especially when the sample consists of very high frequency data, then classical time series statistical tests become almost meaningless and no conclusions may be drawn from them.

### 3.6 Statistical Analysis of 15 Minute ZAR/USD Time Series

#### 3.6.1 Time Series Plots

Figure 3-16 below shows the time plot of the Intraday (15 minute intervals) ZAR/USD exchange rate time series generated by EViews.
Figure 3-16: Time Series Plot of Intraday (15 minute intervals) ZAR/USD Exchange Rate

From Figure 3-16 above, the following observations are made:

1. The observations are similar to that of the 5 minute intraday time series.

2. The observed intraday (15 minute intervals) ZAR/USD time series is not (weakly) stationary.

3. The general trend of this volatile time series is decreasing, i.e., a strengthening of the Rand.

Figure 3-17 below shows the time plot of the intraday (15 minute intervals) ZAR/USD exchange rate returns time series generated by EViews.
From Figure 3-17 above, the following characteristics are noted:

1. The observed intraday (15 minute intervals) returns time series appears relatively stationary.

2. As is the case with the 5 minute intraday time series, it is almost meaningless to comment on the extreme returns occurring in this time series, since the intervals in this intraday time series are 15 minutes. However, it is observed that the 15 minute returns series includes more extremes than the 5 minute returns series.

3.6.2 Test for Stationarity

The unit root test is conducted in EViews to ensure that the daily returns time series is indeed stationary.

Table 3.1 contains the results of the Augmented Dickey - Fuller (ADF) test on the intraday (15 minute interval) ZAR/USD returns time series:

From Table 3.1, it is noted that:
ADF Test Statistic = -55.50416 < -3.4340 (Critical Value at 1%).

Thus, the null hypothesis of a unit root (or non-stationary) time series is rejected at the 1% level, which implies that there is strong evidence to suggest that the observed returns time series is stationary.

3.6.3 Histogram and Descriptive Statistics

Figure 3-18 below shows the histogram for the intraday (15 minute interval) ZAR/USD exchange rate returns time series generated by EViews, and some of the important descriptive statistics of the data:

![Histogram and Descriptive Statistics](image)

Figure 3-18: Histogram and Descriptive Statistics for Intraday (15 minute intervals) ZAR/USD Returns Time Series

From Figure 3-18 above, the following observations can be made:

1. The returns series has a mean return of -0.0000131 i.e. very close to zero.
2. The median return of 0.000000 is slightly greater than the mean return.

3. The standard deviation of the daily returns is 0.001266, implying that the largest negative return (appreciation of the Rand) is 10.849 standard deviations away from the mean daily return and the largest positive return (depreciation of the Rand) is 11.262 standard deviations away from its mean.

4. The range of the observed daily returns time series is:

\[ \text{Range} ([r_t]) = \max \{r_t\} - \min \{r_t\} = 0.014245 - (-0.013748) = 0.027993 (2.799\%) \]

The range indicates that the returns have a relatively small range of values. The standard deviation of the returns is relatively small in comparison to the range, resulting from a large number of returns close in value to the mean return thereby offsetting the extreme positive and negative returns, suggesting that the distribution of the returns time series is highly-peaked at the mean.

5. The coefficient of skewness of -0.003321 > 0 indicates that the daily returns time series distribution is asymmetric about its mean, and is more negatively skewed than the corresponding normal distribution (which has a coefficient of skewness of 0), implying that the distribution has tails that extend further towards the left than to the right due to the occurrence of more extreme negative returns than extreme positive returns.

6. The coefficient of kurtosis of 16.70955 > 3 indicates that the distribution is significantly leptokurtic (i.e. sharp peaked and heavy-tailed as compared to the corresponding normal distribution, which has a coefficient of kurtosis of 3).

7. The Jarque-Bera test statistic value measures the difference of the skewness and kurtosis of the time series from those of the normal distribution. At 115018.7 it is far larger than the critical chi-squared value (\(\chi^2_{0.01,2}\)) of 5.99, which is confirmed by an observed p-value of 0.000000. The null hypothesis of normality is rejected at the 5% (and even at the 1%) significance level. Thus, there is very strong evidence to suggest that the distribution of the daily returns time series is non-Gaussian.

57
3.6.4 Normal Quantile-Quantile Plots

Figure 3-19 below shows the normal Q-Q plot for the observed intraday (15 minute intervals) returns time series computed by EViews:

![Normal Quantile-Quantile Plot](image)

Figure 3-19: Normal Quantile-Quantile Plot of Intraday (15 minute interval) Return Time Series

From Figure 3-19 above, the following observations can be made:

1. The distribution of the returns time series is evidently not normal, as the plot follows a non-linear pattern.

2. The extreme negative quantiles appear to have a slightly higher magnitude than the extreme positive quantiles, indicating that the distribution for the returns will have a slightly longer left tail than the right tail. We would therefore expect the coefficient of skew for the observed daily returns time series to be negative, since the coefficient of skew is sensitive to outlying observations.

3. The Q-Q plot is not continuous at 0 since the return series contains many zeroes.
3.6.5 Autocorrelation and Partial Correlation

Figure 3-20 below illustrates the sample autocorrelation function (acf) and partial autocorrelation function (pacf) of the observed returns time series generated by EViews for lags up to 15 intervals:

<table>
<thead>
<tr>
<th>Sample: 14687</th>
<th>Included observations: 14687</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Autocorrelation</td>
<td>Partial Correlation</td>
</tr>
<tr>
<td>1</td>
<td>-0.142</td>
</tr>
<tr>
<td>2</td>
<td>-0.008</td>
</tr>
<tr>
<td>3</td>
<td>0.015</td>
</tr>
<tr>
<td>4</td>
<td>0.013</td>
</tr>
<tr>
<td>5</td>
<td>-0.012</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>0.007</td>
</tr>
<tr>
<td>8</td>
<td>-0.006</td>
</tr>
<tr>
<td>9</td>
<td>-0.012</td>
</tr>
<tr>
<td>10</td>
<td>-0.000</td>
</tr>
<tr>
<td>11</td>
<td>0.012</td>
</tr>
<tr>
<td>12</td>
<td>-0.012</td>
</tr>
<tr>
<td>13</td>
<td>0.007</td>
</tr>
<tr>
<td>14</td>
<td>0.000</td>
</tr>
<tr>
<td>15</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Figure 3-20: Correlogram for ZAR/USD Intraday (15 minute intervals) Returns Time Series

From Figure 3-20 above, the following interpretations can be made:

1. The lag 1 acf and pacf estimate at 0.142 is far larger than the 95% standard error bounds (as indicated by the dotted line), which implies that the lag 1 acf and pacf estimate is significantly different from zero at the 5% significance level. An acf and pacf value of 0.142 implies that the lag one effect only explains 2.02% (0.142^2 = 0.020164 or 2.02%) of the variation.

2. No evidence of exponential decay exists in the acf as lags increase, implying that the observed returns time series cannot be explained by a pure autoregressive process of order k.
3. The approximate 95% standard error for both the acf and pacf estimates is $\frac{2}{\sqrt{T}}$, where $T$ is the number of observations in the observed time series. The observed returns time series has a sample size of 14687 observations, which is extremely large. As a result, the standard error bound of the estimates will be extremely small ($\pm \frac{2}{\sqrt{14687}} = \pm 0.016503$). Thus, if an acf or pacf estimate of order $k$ has a value exceeding this error bound, it cannot be used as conclusive evidence for significance at order $k$.

4. At lag 1 the Q-statistic has an extremely large value of 296. The reported probabilities (p-values) for the Q-statistics are 0.000 for all lags. Therefore the null hypothesis of no autocorrelation up to order $k$ is rejected at all $k = 1, 2, \cdots, 14687$, and the corresponding acf is significant at the 5% level. The Q-statistic is dependent on the sample size, and the larger the sample size the larger the value of the Q-statistic, which explains why all the reported p-values are zero. Thus, a significant Q-statistic cannot be taken as conclusive evidence for a significant order $k$ acf.

5. Classical time series often fail in the presence of large sample sizes and high frequency samples.

### 3.7 Statistical Analysis of 30 Minute ZAR/USD Time Series

#### 3.7.1 Time Series Plots

Figure 3-21 below shows the time plot of the Intraday (30 minute intervals) ZAR/USD exchange rate time series generated by EVIEWS:
Figure 3-21: Time Series Plot of Intraday (30 minute intervals) ZAR/USD Exchange Rate

From Figure 3-21 above, the following observations are made:

1. The observed intraday (30 minute intervals) ZAR/USD time series is not (weakly) stationary.

2. The general trend of this volatile time series is decreasing, i.e., a strengthening of the Rand.

Figure 3-22 below shows the time plot of the intraday (30 minute intervals) ZAR/USD exchange rate returns time series generated by EViews:
From Figure 3-22 above, the following characteristics are noted:

1. The observed intraday (30 minute intervals) returns time series appears relatively stationary.

2. As is the case with the 5 minute and 15 minute intraday time series, it is not make much sense to comment on the extreme returns occurring in this time series, since the intervals in this high frequency time series are 30 minutes. It is often the case that if the ZAR/USD exchange rate drops by a substantial amount, say 50 cents, it recovers most of the loss by end of day.

3.7.2 Test for Stationarity

The unit root test is conducted in EViews to ensure that the daily returns time series is indeed stationary.

Table 3.5 below contains the results of the Augmented Dickey - Fuller (ADF) test on the intraday (30 minute interval) ZAR/USD exchange rate returns time series:

From Table 3.5 above, it is noted that:
ADF Test Statistic = $-39.47671 < -3.4344$ (Critical Value at 1%).

Thus, the null hypothesis of a unit root (or non-stationary) time series is rejected at the 1% level, which implies that there is strong evidence to suggest that the observed returns time series is stationary.

### 3.7.3 Histogram and Descriptive Statistics

Figure 3-23 below shows the histogram for the intraday (30 minute interval) ZAR/USD exchange rate returns time series generated by EViews and, some of the important descriptive statistics of the data:

![Histogram and Descriptive Statistics](image)

**Figure 3-23: Histogram and Descriptive Statistics for Intraday (30 minute intervals) ZAR/USD Returns Time Series**

From Figure 3-23 above, the following observations can be made:

1. The returns series has a mean return of $-0.000026$ i.e. very close to zero.
2. The median return of 0.000000 is slightly greater than the mean return.

3. The standard deviation of the daily returns is 0.001662. The largest negative return (appreciation of the Rand) is 9.727 standard deviations away from the mean daily return and the largest positive return (depreciation of the Rand) is 8.826 standard deviations away from its mean.

4. The range of the observed daily returns time series is:

\[ \text{Range}(\{r_t\}) = \max \{r_t\} - \min \{r_t\} = 0.014642 - (-0.015141) = 0.030000 \ (3.078\%) \]

Therefore the returns have a relatively small range of values. The standard deviation of the returns is relatively small in comparison to the range, resulting from a large number of returns close in value to the mean return thereby offsetting the extreme positive and negative returns, suggesting that the distribution of the returns time series is highly-peaked at the mean.

5. The coefficient of skewness of \(-0.075628 < 0\) indicates that the daily returns time series distribution is asymmetric about its mean, and is more negatively skewed than the corresponding normal distribution (which has a coefficient of skewness of 0), implying that the distribution has tails that extend further towards the left than to the right due to the occurrence of more extreme negative returns than extreme positive returns.

6. The coefficient of kurtosis of 13.69512 \(\gg 3\) indicates that the distribution is significantly leptokurtic (i.e. sharp peaked and heavy-tailed as compared to the corresponding normal distribution, which has a coefficient of kurtosis of 3).

7. The Jarque-Bera test statistics value measures the difference of the skewness and kurtosis of the time series with those from the normal distribution. At 35004.24 it is far larger than the critical chi-squared value \((\chi^2_{0.05,2})\) of 5.99, which is confirmed by an observed p-value of 0.000000. This implies that null hypothesis of normality is rejected at the 5% (and even at the 1%) significance level. Thus, there is very strong evidence to suggest that the distribution of the daily returns time series is non-Gaussian.
3.7.4 Normal Quantile-Quantile Plots

Figure 3-24 below shows the normal Q-Q plot for the observed intraday (30 minute intervals) returns time series computed by EViews:

From Figure 3-24 above, the following observations can be made:

1. The distribution of the returns time series is evidently not normal, as the plot follows a non-linear pattern. The extreme negative quantiles appear to have a slightly higher magnitude than the extreme positive quantiles, indicating that the distribution for the returns will have a slightly longer left tail than the right tail. We would therefore expect the coefficient of skew for the observed daily returns time series to be negative, since the coefficient of skew is sensitive to outlying observations.

2. The Q-Q plot is not continuous at 0 since the return series contains many zeroes.
3.7.5 Autocorrelation and Partial Correlation

Figure 3-25 below illustrates the sample autocorrelation function (acf) and partial autocorrelation function ( pacf) of the observed returns time series generated by EViews for lags up to 15 intervals:

<table>
<thead>
<tr>
<th>Sample: 1 7343</th>
<th>Included observations: 7343</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autocorrelation</td>
<td>Partial Correlation</td>
</tr>
<tr>
<td>1</td>
<td>1.000</td>
</tr>
<tr>
<td>2</td>
<td>0.018</td>
</tr>
<tr>
<td>3</td>
<td>0.091</td>
</tr>
<tr>
<td>4</td>
<td>-0.011</td>
</tr>
<tr>
<td>5</td>
<td>0.009</td>
</tr>
<tr>
<td>6</td>
<td>-0.005</td>
</tr>
<tr>
<td>7</td>
<td>0.006</td>
</tr>
<tr>
<td>8</td>
<td>0.035</td>
</tr>
<tr>
<td>9</td>
<td>0.002</td>
</tr>
<tr>
<td>10</td>
<td>-0.005</td>
</tr>
<tr>
<td>11</td>
<td>-0.014</td>
</tr>
<tr>
<td>12</td>
<td>0.008</td>
</tr>
<tr>
<td>13</td>
<td>0.011</td>
</tr>
<tr>
<td>14</td>
<td>0.008</td>
</tr>
<tr>
<td>15</td>
<td>0.013</td>
</tr>
</tbody>
</table>

Figure 3-25: Correlogram for ZAR/USD Intraday (30 minute intervals) Returns Time Series

From Figure 3-25 above, the following interpretations can be made:

1. The lag 1 acf and pacf estimate at -0.091 is far larger than the 95% standard error bounds (as indicated by the dotted line), implying that the lag 1 acf and pacf estimate is significantly different from zero at the 5% significance level. An acf and pacf value of -0.091 implies that the lag 1 effect only explains 0.83% (\(-0.091^2 = 0.008281\) or 0.83%) of the variation.

2. No evidence of exponential decay exists in the acf as lags increase, implying that the observed returns time series cannot be explained by a pure autoregressive process of order k. The relatively large autocorrelation at higher lags suggests that long memory
in the time series may exist.

3. The approximate 95% standard error for both the acf and pacf estimates is \( \frac{2}{\sqrt{T}} \), where \( T \) is the number of observations in the observed time series. The observed returns time series has a sample size of 7343 observations, which is extremely large. As a result, the standard error bound of the estimates will be extremely small (\( \pm \frac{2}{\sqrt{7343}} = \pm 0.023340 \)). Thus, if an acf or pacf estimate of order \( k \) has a value exceeding this error bound, it cannot be used as conclusive evidence for significance at order \( k \).

4. At lag 1 the Q-statistic has an extremely large value of 61.009. The reported probabilities (p-values) for the Q-statistics are 0.000 for all lags. Therefore the null hypothesis of no autocorrelation up to order \( k \) is rejected at all \( k = 1, 2, \ldots, 7343 \), and the corresponding acf is significant at the 5% level. The Q-statistic is dependent on the sample size, and the larger the sample size the larger the value of the Q-statistic, which explains why all the reported p-values are zero. Thus, a significant Q-statistic cannot be taken as conclusive evidence for a significant order \( k \) acf.

5. When there is a large sample size, and especially when the sample consists of high frequency data, then classical time series statistical tests become meaningless and no conclusions may be drawn from them.

### 3.8 Statistical Analysis of 60 Minute ZAR/USD Time Series

#### 3.8.1 Time Series Plots

Figure 3-26 below shows the time plot of the Intraday (60 minute intervals) ZAR/USD exchange rate time series generated by EViews:
Figure 3-26: Time Series Plot of Intraday (60 minute intervals) ZAR/USD Exchange Rate

From Figure 3-26 above, the following observations are made:

1. The observed intraday (60 minute intervals) ZAR/USD time series is not (weakly) stationary.

2. The general trend of this volatile time series is decreasing, i.e., a strengthening of the Rand.

Figure 3-27 below shows the time plot of the intraday (60 minute intervals) ZAR/USD exchange rate returns time series generated by EViews.
Figure 3.27: Time Series Plot of Intraday (60 minute intervals) ZAR/USD Returns

From Figure 3.27 above, the following characteristics are noted:

1. The observed intraday (60 minute intervals) returns time series appears relatively stationary.

3.8.2 Test for Stationarity

The unit root test is conducted in EViews to explore whether the daily returns time series is indeed stationary.

Table 3.6 below contains the results of the Augmented Dicky - Fuller (ADF) test on the intraday (60 minute interval) ZAR/USD exchange rate returns time series:

<table>
<thead>
<tr>
<th>ADF Test Statistics</th>
<th>(-26.49337)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacKinnon critical values for rejection of (H_0):</td>
<td>1% Critical Value = (-3.4352)</td>
</tr>
<tr>
<td></td>
<td>5% Critical Value = (-2.6628)</td>
</tr>
<tr>
<td></td>
<td>10% Critical Value = (-2.5673)</td>
</tr>
</tbody>
</table>

Table 3.6: Results from Augmented Dicky-Fuller Test for (60 min) ZAR/USD Returns

From Table 3.6 above, it is noted that:
ADF Test Statistic = \(-26.49357 \ll -3.4352\) (Critical Value at 1%).

Thus, the null hypothesis of a unit root (or non-stationary) time series is rejected at the 1% level. Therefore there is strong evidence to suggest that the observed returns time series is stationary.

### 3.8.3 Histogram and Descriptive Statistics

Figure 3-28 below shows the histogram for the intraday (60 minute interval) ZAR/USD exchange rate returns time series generated by EViews and, some of the important descriptive statistics of the data:

![Histogram and Descriptive Statistics](image)

From Figure 3-28 above, the following observations can be made:

1. The returns series has a mean return of \(-0.000622\) i.e. very close to zero.

2. The median return of 0.000000 is slightly greater than the mean return.

3. The standard deviation of the daily returns is 0.002274. The largest negative return (appreciation of the Rand) is 7.919 standard deviations away from the mean daily return.
4. The range of the observed daily returns time series is:

\[ \text{Range} \{ r_t \} = \max \{ r_t \} - \min \{ r_t \} = 0.014642 - (-0.018059) = 0.032701 \ (3.271\%) \]

The range indicates that the returns have a relatively small range of values. The standard deviation of the returns is relatively small in comparison to the range, resulting from a large number of returns close in value to the mean return thereby offsetting the extreme positive and negative returns, suggesting that the distribution of the returns time series is highly-peaked at the mean.

5. The coefficient of skewness of \(-0.006465 > 0\) indicates that the daily returns time series distribution is asymmetric about its mean, and is more negatively skewed than the corresponding normal distribution (which has a coefficient of skewness of 0). The distribution has tails that extend further towards the left than to the right due to the occurrence of more extreme negative returns than extreme positive returns.

6. The coefficient of kurtosis of 11.05058 \(> 3\) indicates that the distribution is significantly leptokurtic (i.e., sharp peaked and heavy-tailed as compared to the corresponding normal distribution, which has a coefficient of kurtosis of 3).

7. The Jarque-Bera test statistic value measures the difference of the skewness and kurtosis of the time series from those of the normal distribution. At 9913.543 the statistic is far larger than the critical chi-squared value \( (\chi^2_{0.05,2}) \) of 5.99, which is confirmed by an observed p-value of 0.000000. The null hypothesis of normality is rejected at the 5% (and even at the 1%) significance level. Thus, there is very strong evidence to suggest that the distribution of the daily returns time series is non-Gaussian.

3.8.4 Normal Quantile-Quantile Plots

Figure 3-29 below shows the normal Q-Q plot for the observed intraday (60 minute intervals) returns time series computed by EViews.
Figure 3-29: Normal Quantile-Quantile Plot of Intraday (60 minute interval) Returns Time Series

From Figure 3-29 above, the following observations can be made:

1. The distribution of the returns time series is evidently not normal, as the plot follows a non-linear pattern.

2. The extreme negative quantiles appear to have a slightly higher magnitude than the extreme positive quantiles, indicating that the distribution for the returns will have a slightly longer left tail than the right tail. We would therefore expect the coefficient of skew for the observed daily returns time series to be negative, since the coefficient of skew is sensitive to outlying observations.

3. The Q-Q plot is not continuous at 0 since the return series contains many zeroes.

3.8.5 Autocorrelation and Partial Correlation

Figure 3-30 below illustrates the sample autocorrelation function (acf) and partial autocorrelation function (pacf) of the observed returns time series generated by EViews for lags up to 15 intervals:
### Table

<table>
<thead>
<tr>
<th>Autocorrelation</th>
<th>Partial Correlation</th>
<th>AC</th>
<th>PAC</th>
<th>Q-Stat</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.059</td>
<td>12.853</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-0.008</td>
<td>13.089</td>
<td>0.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-0.018</td>
<td>14.252</td>
<td>0.003</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.050</td>
<td>23.345</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.004</td>
<td>23.412</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-0.018</td>
<td>24.564</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-0.011</td>
<td>24.988</td>
<td>0.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-0.001</td>
<td>24.990</td>
<td>0.002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.012</td>
<td>25.524</td>
<td>0.002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-0.021</td>
<td>27.076</td>
<td>0.003</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.003</td>
<td>27.106</td>
<td>0.004</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-0.008</td>
<td>27.351</td>
<td>0.007</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-0.014</td>
<td>28.046</td>
<td>0.009</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-0.006</td>
<td>28.189</td>
<td>0.013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.006</td>
<td>28.326</td>
<td>0.020</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-30: Correlogram for ZAR/USD Intraday (60 minute intervals) Returns Time Series

From Figure 3-30 above, the following interpretations can be made:

1. The lag 1 acf and pacf estimate at $-0.059$ is far larger than the 95% standard error bounds (as indicated by the dotted line), which implies that the lag 1 acf and pacf estimate is significantly different from zero at the 5% significance level. An acf and pacf value of $-0.059$ implies that the lag 1 effect only explains $0.35\%(-0.059^2 = 0.003481$ or $0.35\%)$ of the variation.

2. No evidence of exponential decay exists in the acf as lags increase, implying that the observed returns time series cannot be explained by a pure autoregressive process of order $k$. The relatively large autocorrelation at higher lags suggests that long memory in the time series may exist.

3. The approximate 95% standard error for both the acf and pacf estimates is $\frac{2}{\sqrt{T}}$, where $T$ is the number of observations in the observed time series. The observed returns time series has a sample size of 3671 observations, which is extremely large. As a result, the
standard error bound of the estimates will be extremely small \((\pm \frac{2}{\sqrt{3671}} = \pm 0.033009)\). Thus, if an acf or pacf estimate of order \(k\) has a value exceeding this error bound, it cannot be used as conclusive evidence for significance at order \(k\).

4. Classical time series often fail in the presence of large sample sizes and high frequency samples.

### 3.9 Concluding Remarks

Based on the analyses and results above, the following conclusions may be drawn:

1. The daily ZAR/USD exchange rate has been extremely volatile during the period under study (02/01/1990 - 28/02/2003), and is still volatile to date. In comparison the daily DEM/USD exchange rate has been rather stable during its period under study (02/01/1990 - 31/12/1998).

2. The distribution of the daily ZAR/USD returns is evidently non-normal and skewed to the right. It is skewed to the right because during the period under study the trend of the currency movement is one of weakness. Although the daily DEM/USD returns distribution is not exactly normal, it is relatively close to the normal distribution.

3. The intraday ZAR/USD data also shows patterns of high volatility. For the 7 month period under study (01/07/2002 - 31/01/2003) the rand moved from 10.30 to 8.58 to the dollar. This strengthening of the rand continued during 2003.

4. The distribution of this very high frequency data (5, 15, 30, 60 minute observations) is very non-normal indeed, since the distributions have extremely high peaks at the mean and much heavier tails than the corresponding normal distributions.

Our findings indicate that the ZAR/USD returns, used in this research, do not conform to a normal distribution. We therefore try to fit a stable distribution, which allow for skewness and heavy tails, to the ZAR/USD returns.
Chapter 4

FITTING A STABLE DISTRIBUTION AND R/S ANALYSIS OF FOREX DATA

The first aim of this chapter is to fit a stable distribution to the foreign exchange data. The second aim is to apply the technique of Re-scaled Range analysis (R/S analysis) on the ZAR/USD forex returns data by calculating Hurst exponents and Fractal dimensions.

4.1 Data Set

The FX data utilised in this chapter was obtained from Standard Bank, kindly provided by Anita Last. The data set comprises ZAR/USD daily data sample from 02/01/1990 until 28/02/2003, and the ZAR/USD intraday data sample (5, 15, 30 and 60 minute observations) is from 01/07/02 until 31/01/2003.

4.2 Fitting a Stable Distribution to Foreign Exchange Data

4.2.1 Overview of Stable Distributions

Stable distributions are a class of probability distributions that allow skewness and heavy tails. It is often argued that financial asset returns are the cumulative outcome of a vast
number of pieces of information and individual decisions arriving almost continuously in time [McCulloch(1996), Rachev and Mittnik(2000)]. In the presence of heavy tails it is natural to consider that returns are approximately governed by a stable non-Gaussian distribution. Stable laws were introduced by Paul Levy(1925) during his investigations of the behaviour of independent random variables. Due to the lack of closed form formulas for densities for all but three distributions (Gaussian, Cauchy, Levy), the stable law can most conveniently be described by its characteristic function.

**Definition 7** A random variable \( X \) has a stable distribution if the characteristic function of \( X \) is given by:

\[
E[\exp(i\theta X)] = \begin{cases} 
\exp \left\{ -\gamma |\theta|^\alpha \left[ 1 - i\beta \text{sign}(\theta) \tan \left( \frac{\alpha \pi}{2} \right) + i\delta \theta \right] \right\} & \text{if } \alpha \neq 1 \\
\exp \left\{ -\gamma |\theta| \left[ 1 + i\beta \frac{1}{\alpha} \text{sign}(\theta) \ln|\theta| \right] + i\delta \theta \right\} & \text{if } \alpha = 1
\end{cases}
\]

where:

\( \alpha \in (0, 2) \) is the index of stability or characteristic exponent,

\( \beta \in [-1, 1] \) is the skewness parameter,

\( \gamma \in (0, \infty) \) is the scale parameter,

\( \delta \in (-\infty, \infty) \) is the location parameter, and

\[
\text{sign}(\theta) = \begin{cases} 
-1 & \text{if } \theta < 0 \\
0 & \text{if } \theta = 0 \\
1 & \text{if } \theta > 0
\end{cases}
\]

i.e.

\( X \sim S(\alpha, \beta, \gamma, \delta, 1) \)

For numerical purposes, it is preferable to use a different parameterisation:

**Definition 8** A random variable \( X^0 \) has a stable distribution if the characteristic function of
$X^0$ is given by:

$$
E [\exp (i\theta X^0)] = \begin{cases} 
\exp \left\{ -\gamma^{\alpha |\theta|^\alpha} \left[ 1 + i\beta \text{sign} (\theta) \tan \left( \frac{\pi \theta}{2} \right) \left( (\gamma |\theta|)^{1-\alpha} - 1 \right) \right] + i\delta^0 \theta \right\} & \text{if } \alpha \neq 1 \\
\exp \left\{ -\gamma |\theta| \left[ 1 + i\beta^2 \text{sign} (\theta) (\ln |\theta| + \ln (\gamma)) \right] + i\delta^0 \theta \right\} & \text{if } \alpha = 1
\end{cases},
$$

where:

- $\alpha \in (0, 2]$ is the index of stability or characteristic exponent,
- $\beta \in [-1, 1]$ is the skewness parameter,
- $\gamma \in (0, \infty)$ is the scale parameter,
- $\delta^0 \in (-\infty, \infty)$ is the location parameter.

i.e.

$$X^0 \sim S^0 (\alpha, \beta, \gamma, \delta^0, 0)$$

In the case of the $S^0$ parameterisation the characteristic function and hence the density and distribution function is jointly continuous in all four parameters. The parameters $\alpha$, $\beta$ and $\gamma$ have the same meaning in both the $S$ and $S^0$ parameterisation, while the location parameters of the two representations are related by:

$$\delta = \begin{cases} 
\delta^0 - \beta \left( \tan \frac{\pi \theta}{2} \right) \gamma & \text{if } \alpha \neq 1 \\
\delta^0 - \beta^2 \gamma \ln \gamma & \text{if } \alpha = 1
\end{cases}$$

### 4.2.2 Fitting the Stable Distribution

The four parameters $\alpha, \beta, \gamma, \delta$ of the $S^0$ parameterisation are estimated using the STABLE 3.04 FORTRAN program. Table 4.1 reports the estimated parameters for daily as well as intradaily ZAR/USD returns time series. This STABLE 3.04 FORTRAN program, written by John P. Nolan, calculates the density (pdf), cumulative distribution function (cdf), and quantiles for a general stable distribution. It also performs maximum likelihood estimation of stable parameters and some exploratory data analysis techniques for assessing the fit of a data set.

From Table 4.1 the following is noted:
<table>
<thead>
<tr>
<th>Frequency of Returns</th>
<th>( \hat{\alpha} )</th>
<th>( \hat{\beta} )</th>
<th>( \hat{\gamma} )</th>
<th>( \hat{\delta} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily</td>
<td>1.3572</td>
<td>0.0419</td>
<td>2.6341 \times 10^{-3}</td>
<td>2.0376 \times 10^{-4}</td>
</tr>
<tr>
<td>60 Minutes</td>
<td>1.6200</td>
<td>-0.0185</td>
<td>1.6688 \times 10^{-3}</td>
<td>-7.8445 \times 10^{-5}</td>
</tr>
<tr>
<td>30 Minutes</td>
<td>1.6022</td>
<td>0.0151</td>
<td>1.1965 \times 10^{-3}</td>
<td>-6.28331 \times 10^{-5}</td>
</tr>
<tr>
<td>15 Minutes</td>
<td>1.6523</td>
<td>0.0022</td>
<td>9.4397 \times 10^{-4}</td>
<td>-3.1668 \times 10^{-5}</td>
</tr>
<tr>
<td>5 Minutes</td>
<td>1.6078</td>
<td>-0.0068</td>
<td>6.4592 \times 10^{-4}</td>
<td>-8.1523 \times 10^{-5}</td>
</tr>
</tbody>
</table>

Table 4.1: Results from fitting Stable Distribution to Daily and Intraday ZAR/USD Returns

1. \( \hat{\alpha} \) determines the rate at which the tails of the distribution taper off and is therefore also known as the tail exponent. None of the return distributions are Gaussian since \( \hat{\alpha} \neq 2 \). Levy(1925) has shown that when \( \hat{\alpha} < 2 \) the tails of stable distributions are asymptotically equivalent to a Pareto law. In general the \( p \)th moment of a stable random variable is finite if and only if \( p < \alpha \). Therefore since \( \alpha > 1 \) for all frequencies we conclude the means of the respective distributions exist. For return distributions that are Gaussian second moments exist. Since \( \hat{\alpha} \neq 2 \), the second moments do not exist but sample moments can be calculated.

2. The distribution of daily returns has infinite variance since \( \hat{\alpha} < 2 \). \( \hat{\beta} > 0 \) implies that the distribution is negatively skewed with the median less than the mode. It is shifted slightly to the right of zero i.e. \( \hat{\beta} > 0 \).

3. The distribution of intraday (60 minute) returns has infinite variance since \( \hat{\alpha} < 2 \). It is positively skewed with the median greater than the mode \( (\hat{\beta} < 0) \), and is shifted slightly to the left of zero \( (\hat{\delta} < 0) \).

4. The distribution of intraday (30 minute) returns has infinite variance since \( \hat{\alpha} < 2 \). It is negatively skewed with the median greater than the mode \( (\hat{\beta} > 0) \), but is shifted slightly to the left of zero \( (\hat{\delta} < 0) \).

5. The distribution of intraday (15 minute) returns has infinite variance since \( \hat{\alpha} < 2 \). It is negatively skewed with the median greater than the mode \( (\hat{\beta} > 0) \), and is shifted slightly to the left of zero \( (\hat{\delta} < 0) \).

6. Similarly the distribution of intraday (5 minute) returns has infinite variance since \( \hat{\alpha} < 2 \). It is positively skewed with the median greater than the mode \( (\hat{\beta} < 0) \), and is shifted
slightly to the left of zero ($\hat{\delta} < 0$).

7. The switching of skewness apparent in the intraday data indicates the instability of the estimation.

Diagnostics for Assessing Stability

Modified percent-percent (p-p) plots are used to assess the goodness-of-fit of the stable distribution. Standard p-p plots tend to overstate behaviour around the mode of the distribution (where it has more variation), and understate behaviour near the tails. To eliminate this problem Michael (1983) defines a "stabilised" p-p plot, which involves a transformation making the variance in the p-p plot uniform. As a result the acceptance regions for a p-p plot become parallel lines above and below the diagonal.

Figure 4-1 below shows the p-p and stabilised p-p plots for daily returns time series:

![Probability plots for daily ZAR/USD returns time series](image)

Figure 4-1: Probability plots for daily ZAR/USD returns time series

Based on the figure above it is noted that the stable distribution fits the observed daily return distribution fairly adequately.

Figure 4-2 below shows the p-p and stabilized p-p plots for intraday (60 min intervals) returns time series:
Figure 4-2. Probability plots for intraday (60 min intervals) ZAR/USD returns time series

From the above figure it appears that the fit of the stable distribution is fairly satisfactory with intraday returns. In the case of the corresponding stabilised p-p plot there is a slight deviation from the reference line at the right and left end. The right and left tails are therefore not well fitted by the stable distribution.

Figure 4-3 below shows the p-p and stabilised p-p plots for intraday (30 min intervals) returns time series:
Figure 4-3: Probability plots for intraday (30 min intervals) ZAR/USD returns time series.

It is noted that the stable distribution is a satisfactory fit for the 30 minute returns distribution. As in the case of the 60 minute returns the right and left tails are not well fitted by the stable distribution, as is evident from the deviation of the stabilised p-p plot from the reference line at the right and left end.

Figure 4-4 below shows the p-p and stabilised p-p plots for intraday (15 min intervals) returns time series:
Figure 4-4: Probability plots for intraday (15 min intervals) ZAR/USD returns time series

The figure above indicates that there is slight deviation of the plot around the reference line especially near the centre (mode). The tails are also not very well fit by the stable distribution.

Figure 4-5 below shows the p-p and stabilized p-p plots for intraday (5 min intervals) returns time series:

Figure 4-5: Probability plots for intraday (5 min intervals) ZAR/USD returns time series

From the figure above it is evident that there is some deviation of the p-p plots from
the reference line. This deviation especially near the centre is attributed to the very high frequency data set including a fair amount of zero returns, i.e., during those 5 minute intervals the exchange rate was unchanged. The departure of the stabilised p-p plots from the reference line near the ends indicates that the tails of this distribution are not well fitted by the stable distribution.

4.2.3 Concluding Remarks

1. We confirmed our previous findings that the daily and intraday ZAR/USD returns are non-normally distributed, with skewness and heavy tails.

2. The stable distribution fits the daily ZAR/USD time series fairly well.

3. In the case of intraday data, the stable distribution is a satisfactory approximation for the returns distribution. The right and left tails are not very well fitted by the stable distribution, as indicated by the deviation of the p-p plot from the reference lines at the ends. For very high frequency data, i.e., 5 minute observations, there is also deviation near the centre of the plot, which is due to the large number of zeroes contained in very high frequency data sets.

4. Fitting experience shows that if the number of zeroes exceeds 25% the p-p plots demonstrate non-linear patterns. Therefore a mixture of a discrete probability distribution and a stable distribution may offer a realistic fitting. However, the current fitting (p-p plots and stabilised p-p plots) of the distributions are limited by the existing software, STABLE 3.04 FORTRAN written by John P. Nolan. The original set of analyses was discarded since the intraday data sets included 25% – 50% zeroes, and clearly demonstrated non-linear patterns. Deviations of the p-p plots from the reference lines were observed at the ends and centres of the plots. The deviation of the p-p plots from the reference lines were greatly evident at the centre of the plots, thus forming a zig-zag line. We then decided to clean the intraday data sets by removing zero returns during non-trading periods, i.e., those returns occurring from midnight to seven in the morning. However, the ZAR/USD intraday trading is not strictly from seven in the morning to midnight daily, on occasion there is no trading activity from nine or ten in the evening. Thus, these intraday subsets
used in the stable analyses presented here may also include a fair amount of zero returns.

4.3 R/S Analysis and the Hurst Exponent

4.3.1 Overview of R/S Analysis and the Hurst Exponent

The Hurst exponent occurs in several areas of applied mathematics, including fractals and chaos theory, long memory processes and spectral analysis. Hurst exponent estimation has been applied in areas ranging from biophysics to computer networking. Estimation of the Hurst exponent was originally developed in hydrology. However, the modern techniques for estimating the Hurst exponent come from fractal mathematics.

The Hurst exponent is also directly related to the "fractal dimension", which gives a measure of the roughness of a surface. For example, the fractal dimension has been used to measure the roughness of coastlines. The relationship between the fractal dimension, $D$, and the Hurst exponent, $H$, is

$$D = 2 - H$$

Estimating the Hurst exponent for a data set provides a measure of whether the data is a pure random walk or has underlying trends. A random process with an underlying trend has some degree of autocorrelation. When the autocorrelation has a very long (or mathematically infinite) decay this process is sometimes referred to as a long memory process.

Random walks can be generated from a defined Hurst exponent. If the Hurst exponent is $0.5 < H < 1.0$, the random walk will be a long memory process. Such data sets are sometimes referred to as fractional Brownian motion. Fractional Brownian motion can be generated by a variety of methods, including spectral synthesis using either the Fourier transform or the wavelet transform.

The fractal dimension provides an indication of how rough a surface is. A small Hurst exponent has a higher fractal dimension and a rougher surface. A larger Hurst exponent has a smaller fractional dimension and a smoother surface.

The values of the Hurst exponent range between 0 and 1. A value of 0.5 indicates a true random walk. A random walk has no correlation between any element and a future
element. A Hurst exponent value with $0.5 < H < 1$ indicates “persistent behavior” (positive autocorrelation). A Hurst exponent value $0 < H < 0.5$ implies a time series with “anti-persistent behavior” (negative autocorrelation).

The Hurst exponent is estimated by calculating the average rescaled range over multiple regions of the data. This expected value is calculated over a set of regions (starting with a region size of 8 or 10 data values) and converges on the Hurst exponent power function. If the data set is a random walk, the expected value will be described by a power function with an exponent of 0.5.

A linear regression line through a set of points, composed of the log of $n$ (the size of the regions on which the average rescaled range is calculated) and the log of the average rescaled range over a set of regions of size $n$, is calculated. The slope of regression line is the estimate of the Hurst exponent. This method for estimating the Hurst exponent was developed and analyzed by Benoit Mandelbrot and his co-authors in papers published between 1968 and 1979.

The Hurst exponent applies to data sets that are statistically self-similar. Statistically self-similar means that the statistical properties for the entire data set are the same for sub-sections of the data set. For example, the two halves of the data set have the same statistical properties as the entire data set. The condition is assumed in estimating the Hurst exponent, where the rescaled range is estimated over sections of different sizes.

The R/S statistic is the range of partial sums of deviations of a time series from its mean rate of change, rescaled by its standard deviation. Denoting a series of returns by $r_t$, the average $m$ and biased standard deviation $S$ of the returns from $t = t_0 + 1$ to $t = t_0 + N$ are

$$m(N, t_0) = \frac{1}{N} \sum_{t=t_0+1}^{t_0+N} r_t$$

$$S(N, t_0) = \left\{ \frac{1}{N} \sum_{t=t_0+1}^{t_0+N} [r_t - m(N, t_0)]^2 \right\}^{1/2}$$

The $N$ partial sum of deviations of $r_t$ from its mean and the range of partial sums are then defined as

$$X(N, t_0, \tau) = \sum_{t=t_0+1}^{t_0+\tau} (r_t - m(N, t_0)), \text{ for } 1 \leq \tau \leq N$$

85
\[ R(N, t_0) = \max_{\tau} X(N, t_0, \tau) - \min_{\tau} X(N, t_0, \tau) \]

The R/S statistic for time scale \( N \) is simply the ratio between the average values of \( R(N, t_0) \) and \( S(N, t_0) \):

\[ [R/S](N) = \frac{\sum_{t_0} R(N, t_0)}{\sum_{t_0} S(N, t_0)} \]

Assuming that a scaling law exists for \([R/S](N)\), we can write

\[ [R/S](N) \approx (aN)^H \]

where \( a \) is a constant and \( H \) is referred to as the Hurst exponent.

### 4.3.2 R/S Analysis of ZAR/USD Returns

The R/S analysis is performed on the daily and intraday ZAR/USD return times series, using the software Benoit. Table 4.2 summarises the results from the R/S analysis:

<table>
<thead>
<tr>
<th>Frequency of Returns</th>
<th>Hurst Exponent (H)</th>
<th>Fractal Dimension (D)</th>
<th>Standard Deviation (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily</td>
<td>0.234</td>
<td>1.766</td>
<td>0.0119</td>
</tr>
<tr>
<td>60 Minutes</td>
<td>0.189</td>
<td>1.811</td>
<td>0.018</td>
</tr>
<tr>
<td>30 Minutes</td>
<td>0.195</td>
<td>1.805</td>
<td>0.0297</td>
</tr>
<tr>
<td>15 Minutes</td>
<td>0.236</td>
<td>1.764</td>
<td>0.022</td>
</tr>
<tr>
<td>5 Minutes</td>
<td>0.248</td>
<td>1.752</td>
<td>0.0216</td>
</tr>
</tbody>
</table>

Table 4.2: Results from R/S Analysis of Daily and Intraday ZAR/USD Returns

It is clear from the results obtained that none of the return series follow a random walk, since the corresponding Hurst exponents (H) are not equal to 0.5. The return series exhibit anti-persistence at both daily and intraday frequency with \( 0 < H < 0.5 \). Since the Fractal Dimension (D) = 2 - H, a large Hurst exponent implies a smaller fractal dimension (smoother surface) and vice versa. As the Hurst exponent decreases, the roughness of the surface increases.

Figure 4-6 below shows the log-log plot of the observed versus expected R/S values for the daily ZAR/USD returns time series.
Figure 4-6: Observed versus Expected R/S Values for Daily ZAR/USD Returns.

1. The graph above illustrates both upward and downward bias of the sample R/S values from the expected R/S values. This phenomenon is especially noticeable towards the end of the daily ZAR/USD returns time series which can be attributed to the volatility during this time period.

Figure 4-7 below shows the log-log plot of the observed versus expected R/S values for the Intraday (60 minute observations) ZAR/USD returns time series.
1. From the figure above we observe upward and downward bias of the sample R/S values from the expected R/S values. There also appears to be some discontinuities in the log-log plot. This high frequency time series, where observations are recorded every hour, is very volatile and the discontinuities represent those times when the market changes direction, i.e., strengthening or weakening of ZAR/USD exchange rate.

Figure 4-8 below shows the log-log plot of the observed versus expected R/S values for the Intraday (30 minute observations) ZAR/USD returns time series.
Figure 4-8: Observed versus Expected R/S Values for Intraday (30 min) ZAR/USD Returns.

1. The log-log plot above is similar in pattern to the log-log plot of the observed versus expected R/S values for the 60 minute returns time series. The upward and downward bias as well as the discontinuities occur at similar points in the time series, also due to the volatility in the time series.

Figure 4-9 below shows the log-log plot of the observed versus expected R/S values for the Intraday (15 minute observations) ZAR/USD returns time series.
1. From the figure above we note that as the frequency of the data points increases there seems to be less upward and downward bias from the expected R/S values. There are still a few points of discontinuity where the exchange rate changes direction.

Figure 4-10 below shows the log-log plot of the observed versus expected R/S values for the Intraday (5 minute observations) ZAR/USD returns time series.
Figure 4-10: Observed versus Expected R/S Values for Intraday (5 min) ZAR/USD Returns.

1. The log-log plot above confirms our previous finding that as the frequency of observations increases, upward and downward bias from the expected R/S values decreases. In fact, this log-log plot has slight upward bias followed by substantial downward bias towards the end of the time series.

4.3.3 Concluding Remarks

1. The results from the R/S analysis indicates that none of the ZAR/USD time series follow random walks.

2. Both the daily and intraday returns time series exhibit anti-persistence.
Chapter 5

MULTI-AGENT MODELLING OF SA FOREX MARKET BY NEURAL NETWORKS

The objective of this chapter is to model and possibly forecast short-term ZAR/USD forex returns. Our objective is to attempt to model the South African forex market through a multi-agent neural network approach.

5.1 An Introduction to Neural Networks

The development of neural networks has been motivated right from its inception by the recognition that the brain computes in an entirely different way from the conventional digital computer. It makes sense therefore, to provide a brief introduction to the workings of the human brain to lay the foundation for our understanding of neural networks.

The struggle to understand the brain owes much to the pioneering work of Ramon y Cajal, who introduced the idea of neurons as structural components of the brain (Haykin, 1998). The neuron is typically much slower than the basic building blocks of the conventional digital computer, however the brain makes up for this relatively slow rate of operation by having a truly staggering number of neurons, with massive interconnections between them. The net result is that the brain is a highly complex, non linear and parallel computer that is ultra efficient. To
put this into context we take the simple example of recognising an old friend in a crowded
restaurant, which is a perceptual recognition task that the brain routinely accomplishes in less
than half a second; a task that would in actual fact take the conventional computer days to
accomplish. How then is the brain able to do this?

At birth, a brain has extensive structure and the ability to build up its own rules through
what is usually referred to as experience. Neurons interact via synapses, which are the ele-
mental, structural and functional units that mediate the interactions between neurons. In
short, a synapse is a simple connection that can impose excitation or inhibition, but not both
on a receptive neuron. The brain builds up its rules through the creation of new synaptic
connections between neurons and by the modification of existing synapses. The process, known
as plasticity, enables the brain to adapt to its surrounding environment, which is the essence
of what we need to understand of the brain's functioning - it is a complex interconnection of
neurons, which interact with each other via synaptic connections. The brain is able to "learn"
or adapt to its surroundings through the creation of new synaptic connections or through the
modification of existing synapses.

In its most general form, a neural network is a machine that is designed to model the way
in which the brain performs a particular task or function. The following definition of a neural
network is adapted from Aleksander and Morton (1990).

"A neural network is a massively parallel distributed processor that has a natural propensity
for storing experiential knowledge and making it available for use. It resembles the brain in
two respects:

1. Knowledge is acquired by the network through a learning process.

2. Interneuron connection strengths, known as synaptic weights, are used to store the knowl-
edge."

A learning algorithm is the procedure used to perform the learning process and its function
is to modify the synaptic weights of the network in an orderly fashion, so as to attain a desired
design objective. The modification of synaptic weights provides the traditional method for
the design of neural networks. Neural networks are typically implemented using electronic
components, or simulated in software on a digital computer.
5.1.1 The Benefits of Neural Networks

Thus far it should be evident that the neural network derives its computing power through:

1. Its massively parallel distributed structure.

2. Its ability to learn and generalize.

By generalization, we mean that the network is able to produce reasonable outputs for inputs not encountered during training. These two information processing capabilities make it possible for neural networks to solve large scale complex problems that are currently intractable. Neural networks therefore offer the following useful properties and capabilities.

1. **Non-linearity**: A neuron, as described earlier, is basically a non-linear device. Therefore a neural network of interconnected neurons will also be non-linear. Non-linearity is a highly important property, particularly if the underlying mechanism responsible for the generation of the input signal is inherently non-linear.

2. **Input-Output mapping**: Supervised learning involves the modification of the synaptic weights of a neural network by applying a training set, where each element from the training sample consists of a unique input signal and the corresponding desired response. A random element from the training set is presented to the network and the synaptic weights are modified to minimise the error between the network output and the desired response. The training of the network is repeated for many elements in the set until the network reaches a steady state, where there are no significant changes in the synaptic weights.

3. **Adaptivity**: Neural networks have a built in capability to adapt their synaptic weights to changes in the surrounding environment. In particular, a neural network trained to operate in a particular environment can easily be re-trained to deal with minor changes in the operating environmental conditions. Of particular importance is the fact that a neural network can be designed to change its synaptic weights in real time when operating in a non-stationary environment.
4. **Evidential Response**: In the context of pattern classification, a neural network can be designed to provide information, not only about what particular pattern to select, but also about the confidence in the decision made. This latter information may be used to reject ambiguous patterns should they arise and thereby improve the classification performance of the network.

5. **Contextual information**: Knowledge is effectively represented by the very structure and activation state of the neural network. Every neuron in the network is potentially affected by the global activity of all other neurons in the network. Consequently, contextual information is dealt with naturally by a neural network.

A more explicit description of what is meant by "knowledge" follows:

**Knowledge Representation**

Fischler and Firschein (1987) provide us with the following definition:

"Knowledge refers to stored information or models used by a person or machine to interpret, predict and appropriately respond to the outside world."

The primary characteristics of knowledge representation are twofold:

1. What information is actually made explicit and

2. How the information is physically encoded for subsequent use.

In real world applications of intelligent machines, it can be said that a good solution depends on a good representation of knowledge. A major task for a neural network, is to learn a model of the world or environment in which it is embedded, and to maintain the model sufficiently consistent with the environment so as to achieve its specified goals. In a neural network, knowledge representation of the surrounding environment is defined by the values taken on by the synaptic weights and thresholds of the network. The form of this knowledge representation constitutes the very design of the neural network, thereby holding the key to its performance. The subject of knowledge representation inside an artificial neural network is however, very complicated. Present understanding of this important subject, knowledge representation, is
the weakest link in the understanding of artificial neural networks. There are four common rules of knowledge representation:

**Rule 1:** Similar inputs, from similar classes, should usually produce similar representations inside the network and should, therefore, be classified as belonging to the same category.

**Rule 2:** Items to be categorized as separate classes, should be given widely different representations in the network.

**Rule 3:** If a particular feature is important, then there should be a large number of neurons involved in the representation of the item in the network.

**Rule 4:** Prior information and invariances should be built into the design of a neural network, thereby simplifying the neural network design by not having to learn them.

### 5.1.2 Artificial Intelligence and Neural Networks

Artificial Intelligence attempts to develop algorithms that require machines to perform tasks that apparently require cognition when performed by humans. An artificially intelligent system should be able to perform three tasks:

1. **store knowledge,**
2. **apply the stored knowledge to solve problems and,**
3. **acquire new knowledge through experience.**

We adapt the following ideas from Sage (1990). An artificial intelligence system has three key components: representation, reasoning and learning.

#### Learning

The learning process is of fundamental importance, to any neural network. The ability of a network to learn from its environment and thereby improve its performance, is possibly the most important property of a neural network. A neural network will learn about its environment through an iterative process of adjustments, applied to its synaptic weights and thresholds. Ideally, the network becomes more knowledgeable about its environment after each iteration of the learning process. Learning in the context of neural networks can be defined as follows:
Learning is a process by which the free parameters of a neural network are adapted through a continuing process of stimulation, by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place.

It follows from this definition of the learning process, that the neural network will progress through the following sequence of events:

1. The neural network is stimulated by an environment.
2. The neural network undergoes changes as a result of this stimulation.
3. The neural network responds in a new way to the environment owing to the changes that have occurred in its internal structure.

We can think of learning as two nodes connected by synaptic weight $w$. At time $n$ we can denote the value of synaptic weight $w$ by $w(n)$. At time $n$ an adjustment $\Delta w(n)$ is applied to synaptic weight $w(n)$ and we can write the updated value as:

$$w(n + 1) = w(n) + \Delta w(n)$$

This equation describes the overall effect of events 1 and 2 implicit in the definition of the learning process presented above. The adjustment $\Delta w(n)$ is computed as a result of the stimulation by the environment (event 1) and the updated value $w(n + 1)$ defined as the change made in the network as a result of this stimulation (event 2). Event 3 takes place when the response of the network, operating with the updated set of parameters is re-evaluated.

There are four basic learning algorithms:

1. Error Correction learning
2. Hebbian learning
3. Competitive learning
4. Boltzmann learning

Error correction learning is based primarily on optimum filtering. In contrast, both Hebbian learning and competitive learning are inspired by neurobiological considerations. Boltzmann
learning is based on thermodynamics and information theory. Additionally there are three basic classes of learning paradigms:

1. Supervised learning
2. Reinforcement learning
3. Self-organised or unsupervised learning

Supervised learning is performed under the supervision of an external assistant. Reinforcement learning involves the use of a critic that evolves through a trial and error process. Unsupervised learning is performed in a self-organised manner, where no assistant or critic is required to instruct synaptic changes in the network.

We now proceed to a slightly more in-depth look at the different learning algorithms and learning paradigms.

**Error Correction Learning** Let $d_k(n)$ denote some desired response or target response for neuron $k$ at time $n$. Let the corresponding value of the actual response of this neuron be denoted by $y_k(n)$. $y_k(n)$ is produced in response to input vector $x(n)$ presented to the network in which neuron $k$ is embedded. The input vector $x(n)$ and the desired response $d_k(n)$ for neuron $k$ represent a particular example or iteration presented to the network at time $n$. It is assumed that this example and all other examples presented to the network are generated by an environment that is probabilistic in nature, but the underlying probability distribution is unknown.

Typically, the actual response $y_k(n)$ of neuron $k$ is different from the desired response $d_k(n)$. Hence, we may define an error signal as the difference between the target response and the actual response as shown by:

$$e_k(n) = d_k(n) - y_k(n)$$

The ultimate purpose of error correction learning is to minimise a cost function based on the error signal $e_k(n)$, such that the actual response of each output neuron in the network approaches the target response for that neuron in some statistical sense. Once a cost function is selected, error correction learning is strictly an optimization problem to which the usual tools
may be brought to bear. A criterion commonly used for the cost function is the mean square error criterion, defined as half the mean of the sum of the squared errors:

\[ J = E \left[ \frac{1}{2} \sum_k e_k^2(n) \right] \]  

(5.1)

where \( E \) is the statistical expectation operator, and the summation is over all the neurons, in the output layer of the network. The factor half is used to simplify subsequent derivations, resulting from the minimization of \( J \) with respect to the free parameters of the network. Equation (5.1) above, assumes that the underlying processes are wide sense stationary. Minimisation of the cost function \( J \) with respect to the network parameters leads to the method of gradient descent. The difficulty with this optimisation procedure however, is that it requires knowledge of the statistical characteristics of the underlying processes, which is overcome by settling for an approximate solution to the optimisation problem. Specifically, we use the instantaneous value of the sum of the squared errors as the criterion of interest:

\[ \xi(n) = \frac{1}{2} \sum_k e_k^2(n) \]

The network is then optimised by minimising \( \xi(n) \) with respect to the synaptic weights of the network. Thus, according to the error correction learning rule (delta rule), the adjustment \( \Delta w_{kj}(n) \) made to the synaptic weight \( w_{kj} \), at time \( n \) is given by

\[ \Delta w_{kj}(n) = \eta e_k(n)x_j(n) \]

where \( \eta \) is a positive constant that determines the rate of learning. The adjustment made to a synaptic weight is proportional to the product of the error signal (measured with respect to some desired response at the output of that neuron) and the input signal of the synapse in question.

Error correction learning relies on the error signal \( e_k(n) \) to compute the correction \( \Delta w_{kj}(n) \) applied to the synaptic weight. Error correction learning behaves like a closed feedback system and as such care needs to be taken in the choice of the learning rate so as to ensure stability of the error correction learning process. The learning rate plays a very significant role in the
performance of error correction learning because it affects both the rate of convergence of the
learning, as well as the convergence itself. If \( \eta \) is small, the learning process proceeds smoothly
but it may take a long time for the system to converge to a stable solution. If on the other
hand, \( \eta \) is large, the rate of learning is accelerated but there is a danger that the learning
process may diverge and the system therefore becomes unstable.

A plot of the cost function versus the synaptic weights characterising the neural network,
consists of a multi-dimensional surface referred to as an “error surface”. Depending on the type
of processing units used to construct the neural network, there are two distinct situations that
may arise:

1. The neural network consists entirely of linear processing units, in which case the error
surface is exactly a quadratic function of the weights in the network. This means the
error surface is bowl-shaped with a unique minimum point.

2. The neural network consists of non-linear processing units, in which case the error surface
has a global minimum as well as local minima.

In both cases, the objective of error correction learning is to start from an arbitrary point
on the error surface and then move toward a global minimum, in a step-by-step process. In
the case of a network with only linear processing units, the objective is attainable. In the case
where the network has non-linear processing units, the goal is not always attainable, because
it is possible for the algorithm to get trapped at a local minimum of the error surface and
therefore, never be able to reach a global minimum.

**Boltzmann Learning** The Boltzmann Learning rule is a stochastic learning algorithm de-
erived from information and thermodynamic considerations. In a Boltzmann machine, the
neurons constitute a recurrent structure and they operate in a binary manner, in that they are
either in an on-state or an off-state. The machine is characterized by an energy function, the
value of which is determined by the particular states, occupied by the individual neurons of the
machine as shown by:

\[
E = -\frac{1}{2} \sum_i \sum_j w_{ji} \sigma_i \sigma_j
\]
The machine operates by choosing a neuron at random at some step of the learning process and flipping its state. The neurons of a Boltzmann machine divide into two functional groups: visible and hidden. The visible neurons provide an interface between the network and the environment in which it operates, whereas the hidden neurons always operate freely. There are two modes of operation that can be considered: the "clamped condition", in which the visible neurons are all clamped onto specific states, determined by the environment and the "free-running condition" in which all the neurons are allowed to operate freely.

A distinctive feature of Boltzmann learning is that it only uses locally available observations under two operating conditions, clamped and free-running.

Hebbian Learning  Hebb's (1949) postulate of learning is the oldest and most famous of all learning rules. The following two part rule can be derived as follows:

1. If two neurons on either side of a connection are activated simultaneously, then the strength of that connection is selectively increased.

2. If two neurons on either side of a connection are activated asynchronously, then that connection is selectively weakened or eliminated.

This type of connection is called a Hebbian connection. Hebbian connections use time-dependent and interactive mechanisms to increase connection efficiency. We will not spend a lot of time focusing on Hebbian Learning.

Competitive Learning  In competitive learning, as the name implies, the output neurons of a neural network compete among themselves for being the one to be activated. So, in a neural network based on Hebbian Learning, several output neurons may be activated simultaneously whereas in the case of competitive learning, only one output neuron is activated. There is substantial evidence for competitive learning playing an important role in the formation of topographic maps, in the brain. There are three basic elements to a competitive learning rule:

1. A set of neurons that are all the same, except for some randomly distributed synaptic weights, which therefore respond differently to a given set of input patterns.

2. A limit imposed on the strength of each neuron.
3. A mechanism that permits the neurons to compete for the right to respond to a given subset of inputs. Therefore, only one output neuron or only one neuron per group is active at a time. What this does, is to allow the individual neurons of the network to learn to specialise on sets of similar patterns, thereby becoming feature detectors.

We now proceed to a brief description of the learning paradigms discussed.

**Supervised Learning** An essential ingredient of supervised or active learning is the availability of an external teacher. The teacher should be thought of as having knowledge of the environment that is represented by a set of input-output examples. The environment is however, unknown to the neural network of interest. If the teacher and the neural network are both exposed to a training vector drawn from the environment, the teacher (because of experience) is able to provide the network with a desired response for the training vector. This desired response of course represents the optimum action to be performed by the neural network. The network parameters are adjusted under the influences of a training vector and an error signal. The error signal, as discussed in the section on error correction learning, is the difference between the actual response of the network and the desired response. The adjustment is carried out iteratively in a step-by-step manner with the aim of eventually making the neural network emulate the teacher. The idea is to transfer as much of the knowledge of the environment that is available to the teacher to the neural network. When this condition is reached the teacher is no longer needed and the neural network is able to deal with the environment on its own.

The supervised learning as we have just described is the same as the error correction learning rule described earlier. Once again we may use the mean square error defined as a function of the free parameters of the system as a performance measure. It can be visualised as a multi-dimensional error surface, whose co-ordinates are the free parameters. Any given operation of the system under the teacher’s supervision is represented by a point on the error surface. For the system to learn from the teacher and improve performance, the operating point must move across the error surface to a local or global minimum. Given an algorithm designed to minimise the cost function of interest and given an adequate set of input-output examples as well as enough time permitted to do the training, a supervised learning system is usually able to perform such tasks as pattern classification and function approximation satisfactorily.
A particular disadvantage of supervised learning is that without a teacher, a neural network cannot learn new strategies for particular strategies that are not covered by the set of examples used to train the network. This limitation may however be overcome by reinforcement learning.

**Reinforcement Learning** Reinforcement learning attempts to increase the likelihood of the network producing an action that leads to a satisfactory state of affairs. We adapt the following from Haykin (1998):

"If an action taken by a learning system is followed by a satisfactory state of affairs, then the tendency of the system to produce that particular action is strengthened or reinforced. Otherwise, the tendency of the system to produce that action is weakened."

Consider a learning system interacting with an environment described by a discrete time dynamical process with a finite set of states $X$. At time step $n = 0, 1, 2...$ the environment is in state $x(n)$, where $x(n)$ is an element of $X$. After observing the environmental state $x(n)$ at time step $n$, the learning system performs an action $a(n)$, selected from a finite set of possible actions $A$ that can depend on $x(n)$. The action $a(n)$ affects the environment, causing it to make a transition from state $x(n)$ to a new state $y$ in a manner independent of its past history. Let $p_{xy}(a)$ denote the probability of this state transition, which depends on action $a(n)$. After action $a(n)$ is taken, the learning system received reinforcement $r(n + 1)$, which is determined in some random manner depending on the state $x(n)$ and action $a(n)$. We assume that the sequence of events described herein is allowed to continue for an indefinite number of time steps.

The objective of reinforcement learning is to find a policy for selecting the sequence of actions that is optimal. We restrict ourselves to a stationary policy that specifies actions based on the current state of the environment alone. More precisely, we assume that the probability that the environment makes a transition from state $x(n)$ to $y$ at time $n + 1$, given that is was previously in states $x(0), x(1)...$ and that the corresponding actions $a(0), a(1)...$ were taken, depends entirely on the current state $x(n)$ and action $a(n)$ as shown by:

$$\text{Prob}\{x(n + 1) = y|x(0), a(0); x(1), a(1);...; x(n), a(n)\}$$

$$= \text{Prob}\{x(n + 1) = y|x(n), a(n)\}$$

103
A sequence of states so defined is said to constitute a first order Markov chain with transition probabilities $p_{xy}(a)$. Assuming that the environment is initially in the state $x(0) = x$, a natural measure of the learning systems performance is the evaluation function defined as follows:

$$J(x) = E \left[ \sum_{k=0}^{\infty} \gamma^k r(k+1) | x(0) = x \right]$$

where the expectation operator $E$ is taken with respect to the policy and is used to select actions by the learning system. The summation term inside the expectation is called the cumulative discounted reinforcement. The term $r(k+1)$ is the reinforcement received from the environment after action $a(k)$ is taken by the learning system; it can be positive (reward), negative (punishment) or zero. The factor $\gamma$ is called the discount rate parameter, whose value lies in the range $0 \leq \gamma \leq 1$. By adjusting $\gamma$ we are able to control the extent to which the learning system is concerned with long term versus short term consequences of actions. The basic idea of reinforcement learning is to learn the evaluation function $J(x)$, so as to predict the cumulative discounted reinforcement to be received in the future.

**Unsupervised Learning** Unsupervised learning does not have an external teacher or critic to supervise the learning process. So there are no specific examples of the function to be learned by the network. Rather, provision is made for a task independent measure of the quality of representation that the network is required to learn. The free parameters of the network are optimised according to this measure.

To perform unsupervised learning, we may use a competitive learning rule. For example, we may use a neural network with two layers, an input layer and an output layer (a competitive layer). The input layer receives the input data and the output layer consists of neurons that compete with each other for the opportunity to respond to features contained in the input data.

**Comparing Supervised and Unsupervised Learning** Among the algorithms used to perform supervised learning, the back propagation algorithm has emerged as the most widely used and successful algorithm for the design of multi-layer feedforward networks. There are two distinct phases to the operation of backpropagation learning, the forward phase and the backward phase. In the forward phase the input signals propagate layer by layer through the
network, eventually producing a response at the output of the network. The actual response is compared to a target response, which produces an error signal that is propagated backwards through the network. In the backward phase the weights are adjusted so as to minimise the sum of the squared errors. Supervised learning algorithms are limited by their poor scaling behaviour, which is illustrated by a multi-layer feedforward network consisting of \( N \) layers. The effect of a synaptic weight in the first layer of the network on the output is dependent on its interactions with approximately \( F_i^N \) other synaptic weights where \( F_i \) is the fan in, defined as the average number of incoming links of neurons in the network. As the size of the network increases, the network will become more computationally intensive. The time required to train the network increases exponentially and becomes too slow.

A possible solution to this scaling problem is to use an unsupervised learning procedure. In particular, if we are able to apply a self organising process in a sequential manner, one layer at a time, it is feasible to train dependent networks in time that is linear in the number of layers. Moreover, with the ability of the self organising network to form internal representations that model the underlying structure of the input data in a more explicit or simple form, it is hoped that the transformed version of the sensory input would be easier to interpret, so that the correct responses could be associated with the networks internal representations of the environment more quickly. The bottom line is that the hybrid use of unsupervised and supervised learning procedures may provide a more acceptable solution than supervised learning alone.

Having briefly described some learning rules and learning paradigms we now proceed to give a brief introduction to some of the learning tasks that neural networks may be employed to solve.

Learning Tasks

The choice of a particular learning procedure is highly influenced by the learning task which the neural network is required to perform. We can identify the following learning tasks:

1. **Approximation** - We are given a non linear input output mapping described by the functional relationship:

   \[ d = g(x) \]
where the vector $\mathbf{x}$ is the input and the scalar $d$ is the output. The function $g(\cdot)$ is assumed to be unknown. The idea would then be to design a neural network that approximates the non-linear function $g(\cdot)$, given a set of examples denoted by the input-output pairs $(\mathbf{x}_1, d_1), (\mathbf{x}_2, d_2), \ldots, (\mathbf{x}_n, d_n)$. This type of approximation problem, described here, would be well suited to supervised learning with $\mathbf{x}$ serving as the input vector and $d$ serving as the desired response.

2. Association - This learning task may take one of two forms namely: autoassociation or heteroassociation. In autoassociation, a neural network is required to store a set of patterns by repeatedly presenting them to the network. Subsequently, the network is presented a partial description or distorted version of an original pattern stored in it, and the task is to retrieve that particular pattern. Heteroassociation differs from autoassociation, in that an arbitrary set of input patterns are paired with another arbitrary set of output patterns. Autoassociation involves the use of unsupervised learning, whereas the type of learning involved in heteroassociation, is of a supervised nature.

3. Pattern Classification - In this type of learning task, there is a fixed number of categories into which stimuli are to be classified. To resolve the task, the neural network first undergoes training during which the network is repeatedly presented a set of input patterns along with the category to which each particular pattern belongs. A new pattern is then presented to the network, which has not been seen before but which belongs to the same population of patterns used to train the network. The task for the neural network is to classify the new pattern correctly, which is a supervised learning problem.

4. Prediction - The issue of predicting is one of the most basic and pervasive learning tasks. Given a set of $M$ past samples $x(n-1), x(n-2), \ldots, x(n-M)$ that are usually uniformly spaced in time, the requirement is to predict the present sample $x(n)$. Prediction may be solved using error correction learning in an unsupervised manner, in the sense that the training examples are drawn directly from the time series itself. Prediction can be viewed as a form of model building, in the sense that the smaller we make the prediction error in a statistical sense, the better the network will serve as a physical model of the underlying stochastic process which is responsible for the generation of the time series.
Other learning tasks that are not described here, include control and beamforming. Here neural networks may be used to control a process and in the case of beamforming, a kind of spatial filtering, the purpose of which is to locate a target signal embedded in a background of additive interference. The diversity of these learning tasks is a testimony to the universality of neural networks.

One last area of mention with regard to the learning process is the statistical nature of the learning process which we proceed to describe now.

**The Statistical Nature of the Learning Process.** We adapt the following description of the statistical nature of the learning process from Haykin (1987). The learning process experienced by a neural network is a stochastic process. The reason for stochasticity is rooted in the environment in which the network is embedded. A neural network is simply one form in which knowledge about a physical environment of interest may be encoded. We consider the example of a physical environment described by a vector $x$ which represents a set of independent variables, and a scalar $d$ representing a dependent variable. Assume that there are $N$ observations of $x$ denoted by $x_1, \ldots, x_N$ and a corresponding set of observations $d$, denoted by $d_1, \ldots, d_N$.

Since we do not have knowledge of the exact functional relationship between $x$ and $d$ we write the relationship as:

$$d = g(x) + \epsilon$$

where $g(x)$ is some function of the argument vector $x$, and $\epsilon$ is a random expectational error. The statistical model described above is called a regressive model. We can define the function $g(x)$ as

$$g(x) = E[d|x]$$

where $E$ is the statistical expectation operator. The conditional expectation $E[d|x]$ defines the value of $d$ that will be realised, on average, given a particular realisation of $x$. Whenever this conditional expectation exists, it can be represented solely as a function of $x$. A special case of the regressive model has two useful properties:
1. The average value of the expectational error $\epsilon$ given any realisation of $x$ is zero, that is

$$E[\epsilon|x] = 0$$

2. The expectational error $\epsilon$ is uncorrelated with the function $g(x)$; that is

$$E[\epsilon g(x)] = 0$$

This property is a well known principle of orthogonality; it simply states that all the information available to us about $x$ has been encoded into the regression function $g(x)$.

The regressive model that we have described is a “mathematical” model, the purpose of which is to use $x$ in order to predict $d$. A neural network provides a “physical” device for implementing this objective. It does so by encoding the knowledge represented by the training data set $\{x_i, d_i | i = 1, 2, ..., N\}$ into a set of synaptic weights. In this context, $x$ represents the input vector and $d_i$ represents the corresponding value of the desired response. Let $w$ denote the synaptic weight vector of the neural network selected to “approximate” the regressive model described above. Let the actual response of the network be defined by:

$$y = F(x, w)$$

The synaptic weight vector $w$ of the network is adjusted in an iterative fashion in response to the error signal $e$, defined as the difference between the desired response $d$ and the actual response of the network $y$, that is

$$e = d - y$$

### 5.2 Multi-Agent Modelling of a Foreign Exchange Market

Having given an introduction to the concept of neural networks we now move on to provide a description of the system we are attempting to model with our neural network.

We may think of a market as a collection of many agents or traders. Each of these traders
is attempting to maximize his/her profit and the market is basically driven by a superposition of the agents' decisions. On a micro level the decisions of the traders will determine the excess supply or demand that exists in the market, and on the macro economic level will determine the price dynamics. Our objective is to attempt to model the South African forex market through a multi-agent neural network approach. Neural networks allow for the fitting of high dimensional non linear models, which are often used in econometrics. A neuron, as we have earlier described it, can actually be interpreted as a decision making model and this property strengthens the relationship to economics. If we are able to think of a neuron as a decision making model then a neural network in fact models the interaction of many decisions, and as such can be interpreted as a market mechanism, which forms the basis for our model.

A market can be thought of as a large number of interacting traders or agents, or in fact as a network of interacting agents. Each trader has access to a wealth of information such as fundamental and technical indicators, news flows and market levels. The trader must evaluate this information and try to maximise profit by anticipating prospective changes to the market price level. We can take the buying and selling decisions of all the traders in the market and sum this activity up to arrive at an excess supply or demand condition. The market clearing condition (where supply equals demand) is not guaranteed, purely because the traders make decisions independently of each other. The imbalances in supply and demand, i.e. excess supply or demand can be corrected by a shift in the market price level. Thus the market is driven by a superposition of trading decisions. This principle of market mechanics can be applied to any market, no matter what the traded asset may be. It follows then, that to explain or predict market prices requires an analysis of the trading decisions and interactions of the traders in the market.

We adapt the following descriptions of market models from Zimmermann, Neuneier and Grothmann (1997). Recent literature shows us many different methods of attempting to model a financial market. Most of these multi-agent approaches however make use of markets simulated under laboratory conditions. The objectives being to study the behaviour of agents, the evolution of trading strategies or the learning of agents. These type of models are able to produce artificial time series as outputs which have the characteristics of real world price dynamics, but most of these models are unable to fit the real world behaviour of financial
markets.

Lux and Marchesi (1999) set up a stochastic simulation model of an artificial financial market where noise traders and fundamentalists interact. Fundamentalists rely on the efficient market hypothesis, they believe that markets are efficient and assets are evaluated based on their fundamental value. Noise traders on the other hand rely on identifying obvious price trends using charts. Noise traders are therefore classified into two groups, optimists who believe in rising market trends and pessimists who believe that the market will fall. There are three basic elements to the multi-agent simulation, i.e., movements of individuals from one group to another, exogenous changes of the fundamental asset value, and endogenous price changes resulting from agents trading decisions. Price adjustments are endogenous responses from the market in order to balance supply and demand. This type of simulation allows for the production of artificial time series that have the same characteristics as real world empirical data.

We now proceed to provide a description of the modelling of a market structure through the use of neural networks. The mathematical theory of neural networks is suited to market modelling primarily because of the ability of networks to approximate any functional structure. A further and important advantage of neural networks is that they allow for the construction of models that are also able to handle high dimensional and non-linear interactions.

A single neuron can be interpreted as an elementary economic decision making process, because it unites the process of information filtering, the formation of a superposition of information and the execution of an action. We can therefore think of the interaction of hundreds of neurons to be similar to the interactions of traders in a market process.

As mentioned above, the agents of a market will make decisions independently from each other with the aim of maximizing their profit. There is no certainty as to whether the market clearing condition will be met or not, in other words we do not know if the market is in an equilibrium state or if there is an excess supply or demand. Whatever the resulting condition, it is determined by an aggregation of the agents buy and sell decisions. In both cases of a market disequilibrium there will be an impact on the market price level. In the case of excess demand, a price increase will cause more agents to be willing to sell goods at the higher price, which will increase the supply, while at the same time, some of the agents who wanted to buy
will not want to do so anymore because of the price increase, which decreases demand. Similarly, the condition of excess supply is corrected by lowering the market price.

The objective of this chapter is to model ZAR/USD forex returns. As we mentioned earlier, a currency can be treated like any asset and as such we can attempt to model it with the multi-agent mechanism described above. Agents participating in a foreign exchange market try to expand their capital (maximise their profit) by forecasting changes in the exchange rate. So, examining the ZAR/USD forex market, agents who believe in ZAR devaluation will allocate their funds to USD, while agents predicting devaluation of the USD will allocate funds to ZAR.

We now try to model a single agent's decision making behaviour in trying to maximise profit. To model the trading behaviour of an agent we need to make a few assumptions. Our first assumption is that each agent has two accounts that reflect his position in ZAR and USD respectively. We can let $x$ denote the balance of the one account in USD at time $t$. Similarly, $y$ is the balance of the second account in ZAR. Any dealing by the agent in either of these currencies will have an impact on both these accounts. For example, if the agent decides to buy 100 USD, the equivalent amount of ZAR must first be subtracted from his ZAR account and 100 USD is credited to the USD account. Using the above as an example, we can model the trading behaviour of an agent by two equations reflecting the upcoming balances of the two accounts. Assuming $p$ is the exchange rate and $z$ the USD amount purchased, we obtain:

\[
x_{t+1}(USD) = x_t(USD) + z_t
\]

\[
y_{t+1}(ZAR) = y_t(ZAR) - p_zt
\]

This set of equations describes very basically the trading mechanism of an agent. We also make the simplifying assumption of no trading costs. Each trader has a decision to make in every time period $t$. In order to solve this decision problem, we need to consider an elementary decision making process. We can break this process up into three steps:

1. Information Filtering.
3. Perform a trading activity.

We examine this process more closely in the context of a trader having to make a decision.
The rapid advancement of technology has drastically improved information flow in the market place and consequently traders are overwhelmed with a wealth of information from which they must sift out those facts which they deem to be relevant to the related foreign exchange rate. An agent who is a technical analyst would focus more of his attention on technical market descriptors and tend to neglect fundamental indicators. The first stage of the decision making process is referred to as the information filtering phase. In the second stage the agent has to evaluate the information that he has collected and form a view on the future development of the observed FX market. Naturally, the evaluation of the data will lead directly to the third phase of the decision making process which is taking action. For example, if after the information filtering and evaluation an agent expects the devaluation of the ZAR, a good strategy and appropriate action to take would be to buy USD.

This elementary decision making process that we have just described can be modelled by a single neuron, which precisely reflects the decision making process. We can illustrate the similarities of the three stages of the decision making process and a single neuron through the following diagram extracted from Zimmermann, Neuneier and Grothmann (Multi-Agent FX-Market Modelling by Neural Networks).

![Diagram of a neuron](image)

Figure 5-1: Basic mathematical model of an artificial neuron.

The input signals \((u_1, u_2, u_3, \ldots u_n)\) are analogous to the market information. They are weighted by factors \((w_1, w_2, w_3, \ldots w_n)\) which represent the information filtering process. The agent is able to emphasise the importance or significance of a particular piece of information by adjusting the corresponding weight higher or lower (higher weights signify more emphasis on a piece of information). In fact information that is deemed to be insignificant can be ignored by setting the corresponding weight factors equal to zero. By adding up the input signals and comparing the sum to a threshold value, the agent is able to evaluate the information and
form an opinion of the future development of the related FX market. Thereafter the result is transferred through a non linearity in order to generate the neurons output signal. The output of the neuron represents precisely the trading activity of the particular agent on the applicable foreign exchange market.

If we understand a single neuron to function in this way then it follows that a neural network that consists of hundreds of such neurons can be seen as an interaction model between a large number of decision makers or traders and hence can be thought of as a market process.

The decision making process is one part of the agents behaviour. We also have to make assumptions about the agents objective function which guides their behaviour. We assume that the trading activities of the agents are driven by a profit maximisation task:

\[ \frac{1}{T} \sum_t \ln \left( \frac{P_{t+1}}{P_t} \right) z_t \]

Each agent will try to expand his net capital base by utilising upcoming FX rate shifts, which is done by maximising the average profit gained over all time periods. The other assumption that we have made thus far is that the agents all act independently of each other. In other words, the trading activities of agent a have no impact on the buying or selling decisions of the other agents in the market. To enable the agents to interact a market clearing condition needs to be included.

\[ \sum z_t = 0 \quad \text{for} \quad t = 1, ..., T \]

The implication is that all buying and selling decisions must sum to zero. We can illustrate more clearly with an example. Assume that there are only two agents in the market, i.e., a and b. The market clearance condition tells us that if agent a wants to buy ZAR he can only do so if agent b is willing to sell an equal amount of ZAR, else the buying and selling decisions in the market will not sum to zero and the market clearance condition will not be met. So, if both agents independently decide to buy ZAR, no trading will take place because there is no supply to balance this demand. Thus the agents' plans are indirectly associated with each other through the market clearance condition. It also follows that the larger the market, the higher the probability that a single agent will complete his buying or selling decisions due to
the plethora of activities taking place.

There are two market paradigms which we may consider in multi-agent modelling of forex markets, explicit dynamics or implicit dynamics. For our study we have chosen to model the ZAR/USD forex market using the explicit market price paradigm.

5.2.1 Explicit Dynamics of a Foreign Exchange Market

Earlier we described the market clearance condition where the amount of a currency desired by the buyers is exactly matched by the amount of currency offered by the sellers. Then all agents in the market are able to complete their desired trades. This market clearance condition may also be interpreted as an economic equilibrium for the foreign exchange market. In equilibrium the supply and demand in the market are exactly matched. If however, the market clearance condition does not hold, then the market is in a disequilibrium state and thus is a consequence of the trading decisions in the market not being matched exactly. One of two conditions then exists - there is either an excess demand or an excess supply. Now, to bring an unbalanced market back into equilibrium, adjustments to the market price are required, which leads to the definition of the explicit market price paradigm, which states that it is the excess supply or demand that causes the market price changes. Thus the market price is influenced by agents individual trading activities, because their trading activities constitute the excess supply or demand.

In the case of market equilibrium, when buying and selling decisions match exactly, there is actually no need to adjust the market price

\[ \sum z_t = 0 \implies \ln \left( \frac{P_{t+1}}{P_t} \right) = 0 \]

Alternatively if the market clearance condition is not met, then an excess supply or demand exists in the market and a price adjustment is required to bring the market back into equilibrium. Let us assume that \( c \) is a proportional constant, then the required foreign exchange rate shift can be derived by

\[ \sum z_t \neq 0 \implies \ln \left( \frac{P_{t+1}}{P_t} \right) = c \sum z_t \]
Basically the price shift required to bring the market back into equilibrium can be estimated by scaling the excess supply or demand by a positive constant, i.e., \( c > 0 \). If for example, there is an excess supply in the market then total planned sales exceed total planned purchases. Therefore the sum of all planned purchases and sales leads to a negative sum (purchases are positive and sales are negative). Since the supply of the currency is greater than its demand, we can probably expect a devaluation in the currency. This devaluation would then lead to a declining exchange rate which effectively means a negative price shift. This negative price shift required to bring the market back into equilibrium can be computed, as illustrated above, by scaling the negative sum of the purchases and sales by a positive constant \( c \) (the resulting price shift will be negative since \( c \) is positive and the sum of the buy and sell decisions is negative).

Before we proceed to describe the implementation of the explicit market price paradigm by neural networks, we give a very brief introduction to the package used in this dissertation to implement neural networks - SENN.

SENN was developed by the Siemens corporation and provides an environment for the modelling of neural networks which may be used for forecasting and classification. SENN also contains tools for the financial analytical interpretation of the network outputs and for the solution of general complex problems.

The package supports various network types as well as multi-faceted learning and optimisation algorithms, thus making it an ideal tool for the forecasting needs of insurance and financial institutions. SENN also provides some much needed analysis tools where conventional systems usually fail. Sensitivity analysis, post-processing procedures and performance analysis are but a few of the analysis functions provided. Additionally several specially developed learning algorithms are provided.

We now proceed to give more details into the modelling of the explicit market price paradigm using a neural network in the SENN environment.
Figure 5-2: SENN Model Topology

Figure (5-2) above, denotes a four layer feed forward network architecture for modelling the explicit market price paradigm. The first layer which is the input cluster contains 12 input nodes referring to 12 input signals (This is referred to as mlp.input[12] in SENN). The second layer is the agents cluster (This is referred to as mlp.agents[100] in SENN). This layer models 100 agents in the foreign exchange market, each with a separate neuron with a $tanh(.)$ activation function. The agents layer therefore consists of 100 neurons representing 100 agents participating in the market. The summing up of the agents buying and selling decisions result in an excess supply or demand (This is referred to as mlp.excessDemand[1] in SENN). The excess supply or demand can be corrected by a shift in the market price level (This is referred to as mlp.price[1] in SENN). The behaviour of a single agent is guided by a profit maximisation error function.

Appendix A presents the relevant SENN codes for the multi-agent modelling of the ZAR/USD returns, that is the topology file (model.top) and the specification file (model.spc).
Use of the Activation Function

As in biological systems, the neural network output function is activated by a transfer function or activation function. A major advancement to artificial neural networks came with the introduction of the activation function, which enables approximation of non-linear relationships between inputs and outputs. The most frequently used functions are the sigmoid function and the hyperbolic tangent function. The application of the activation function has been particularly useful in financial and economic analysis, since most financial and economic relationships are non-linear functions. Furthermore, it is difficult to have a priori knowledge of the functional form of economic relationships. An incorrect functional form of the model leads to incorrect estimates of parameters.

The sigmoid function, \( f(x) = \frac{1 + e^{-x}}{1 - e^{-x}} \), is bounded between 0 and 1. The hyperbolic tangent function, \( f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \), has the same structure as the sigmoid function, but it is bounded between -1 and 1. It is often used when dealing with variables that may take on negative values. Figure 5-3 below graphically illustrates these activation functions.

![Figure 5-3: Tanh and Sigmoid Activation Functions](image)

The matrix of weights that connects the inputs layer to the agents layer is a sparse connector, implying that many of its weights are equal to zero which is very important for two reasons. Firstly, it assists in the information filtering step in the decision making process earlier, i.e., it does not give all the agents access to all the available information. Secondly, it helps to guarantee a wide range of trading strategies, by ensuring that different agents get access to different subsets of the input information. Since real agents are heterogeneous we must ensure
that the available information is widely distributed amongst them.

We have to aggregate the agents decisions in the market to determine the aggregate supply or demand in the market. To enable our model to calculate the excess supply or demand we connect the agents cluster to the excess demand cluster by a fixed matrix. The shift in the FX rate is then estimated by the last layer, i.e., the market price layer.

We now proceed to provide some explanations for the various inputs and setup used for our multi-agent model, followed by the results and conclusions.

**Input Data**

The input data to the model is the information that is provided to the agents and on which they will base their trading decisions. It is therefore important that an appropriate input data set is provided to the model. We are guided by Zimmermann, Grothmann and Neuneier's basic idea that every major currency is driven by at least four major influences. These are the development of:

1. other major currencies
2. the national and international bond market
3. stock prices
4. commodity prices

In choosing the input data for our model we needed to consider the availability of historical time series for the different assets. We have chosen to use the following set of data (from 02/01/1990 until 31/12/2002) as input to model the daily ZAR/USD exchange rate.

1. Daily British Pound/US Dollar (GBP/USD) exchange rate
2. Daily Swiss Franc/US Dollar (CHF/USD) exchange rate
3. Daily Deutsche Mark/US Dollar (DEM/USD) exchange rate
4. Daily British Pound/SA Rand (GBP/ZAR) exchange rate
5. The Gold price in dollars

118
6. The JSE All Share Index

Due to the fact that most of the data in our time series is for the period before the introduction of the Euro, we use some of the currencies prior to the introduction of the Euro such as the DEM. Other currencies that we choose to use are the GBP and the CHF. We have based our analysis on the assumptions that many of the underlying dynamics of the major economies such as the US Economy and the economies of the other respective countries affect the ZAR/USD exchange rate. In addition we have also used the dollar gold price as we believe that depending the trading strategies employed, gold is sometimes seen as a safe haven asset and in such cases may have a relationship to dollar weakness. In other words we believe that the dollar gold price also encapsulated some important dynamics of the foreign exchange market. With the South African market having a large rand hedge component, the major stock market index is sometimes impacted significantly by rand strength or weakness. For this reason we include a time series of the historic stock market level (as represented by the JSE All Share Index).

Other input series that we may consider including are the yields on the benchmark R153 government bond. In high interest rate environments South African bonds would have attracted a great deal of off shore investment and as such, the price dynamics of the bond yields would also encapsulate some information on the supply and demand for the Rand and possibly the Dollar. For similar reasons we may also consider using the yields on the 10 year US Treasury Note. However, as mentioned earlier, lengthy historical time series for these securities are not readily available. Our hope is therefore that the six input series that we have chosen to use encapsulates significant information of the dynamics of the ZAR/USD foreign exchange market, for the model to learn from. Before proceeding to describe preprocessing, we briefly mention our split of the data into a training set, a validation set and a generalisation set. The training set contains a period of data that is presented to the model to facilitate learning. The validation set contains some data from the training set as well as some data that is not present in the training set and allows us to test the models response to data patterns not previously encountered in the training set. The generalisation set contains data that is neither in the training set nor the validation set and is used to evaluate the models performance once training is complete. We now define the periods in our data set that were used for training, validation and generalisation in Table(5.1).
<table>
<thead>
<tr>
<th>Set</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>03/01/1990 to 03/09/1996</td>
</tr>
<tr>
<td>Validation</td>
<td>03/09/1995 to 03/09/1996</td>
</tr>
<tr>
<td>Generalisation</td>
<td>03/09/1996 to 31/12/2002</td>
</tr>
</tbody>
</table>

Table 5.1: Data subsets for Training, Validation and Generalisation

Our analysis also used a generalisation set for the period 03/09/1996 to 29/12/2000 which excluded the high volatility spike caused by severe depreciation of the rand during 2001. We now proceed to describe the process of preprocessing data before presenting it to the neural network.

**Preprocessing** It is often believed that the characteristics of the available data determine the quality of a neural network model. We learn however from Zimmermann, Grothman and Neuneier (1997) that this can be a misleading view, more so if the amount of useful information that can be extracted from the data is small.

Preprocessing of the data is firstly required, obviously for scaling. We would like to transform the different time series such that each series has a mean value of zero and a statistical variance of one. Various preprocessing functions may be used and in the field of financial forecasting, these functions are often derived from technical analysis in order to capture some of the underlying dynamics of the financial markets. We have chosen to use the following simple transformations adapted from research done by Zimmermann, Grothmann and Neuneier.

If the original time series which has been selected as an input, is changing very slowly with respect to the prediction horizon, i.e., there is no clearly identifiable mean reverting equilibrium, then an indicator for the inertia and information concerning the driving force has been proven to be very informative. The inertia can be described by a momentum and the force by the acceleration of the time series. If we have a prediction horizon of \( n \) steps into the future the original time series \( x_t \) is transformed in the following way:

\[
\text{momentum: } \hat{x}_t = \text{scale} \left( \frac{x_t - x_{t-n}}{x_{t-n}} \right) \\
\text{force: } \tilde{x}_t = \text{scale} \left( \frac{x_t - 2x_{t-n} + x_{t-2n}}{x_{t-n}} \right)
\]

120
In the momentum equation the relative difference is computed to eliminate exponential trends which, for example, may be caused by inflationary influences. Using only the pre-processing functions of the momentum equation typically leads to poor models which only follow obvious trends. The forces, i.e., the transformations using the force equation are important to characterize the turning points of the time series.

Sometimes though, a time series may be fast in returning to its equilibrium state after new information has entered the market, as is the case for most prices of goods and stock rates. In this case, we substitute the force equation by a description of the forces which drive the price back to the estimated equilibrium. A simple way to estimate the underlying price equilibrium is to take the average over some past values of the time series. Instead of using the relative difference between the estimate and the current value, we look at the difference between the equilibrium and the past value, which lies in the middle of the averaging window, which can be formulated as

\[ \hat{x}_t = \text{scale} \left( \frac{x_t-n - \frac{1}{2n+1} \sum_{r=0}^{2n} x_{t-r}}{x_{t-n}} \right) \]

In preprocessing our input data we chose to use the transformations describe by the momentum and force equations, resulting in the original 6 inputs being transformed into 12 inputs.

The following explanation details some of the specifics that had to be considered in setting up the neural network model. Firstly we have to make sure that the agents do not have access to all the information available, which as we discussed earlier is important for two reasons. Firstly it assists in the information filtering step in the decision making process discussed earlier, i.e., it does not give all the agents access to all the information available. Secondly it helps to guarantee a wide range of trading strategies, by ensuring that different agents get access to different subsets of the input information. The connector from the input layer to the agents layer must be a sparse connector, meaning that most of its weights will be set to zero, which is accomplished in our model through the use of weight pruning. Weight pruning is a method of limiting the weights in a connector. Our aim is to remove roughly 60% of the weights from the connector. SENN provides us with the functionality to do this with weight pruning which randomly prunes weights from the connector.

Next we have to ensure that only the sum of the agents buying and selling decisions is passed
through to the excess demand layer of the network. Thus, we have to freeze the connector from the agents layer to the excess demand layer and initialise their values to 0.1. Once again SENN provides the functionality to implement this constraint. Freezing the connector means that its free parameters of weights will not be changed during the learning process. It is important that only the sum of the agents buy and selling decisions is passed to the next level as the explicit market paradigm dictates that the price shift is determined by the excess supply or demand in the market which is in turn determined by the sum of the agents buy or sell decisions.

Finally we need to ensure that the sum of the agents trades, which is fed to the output of the model (the market price shift), is scaled to appropriate level that fits the market price. For this reason only positive values are allowed in the connector that connects the excess demand layer to the market price layer. The scaling parameter that is used to bring the agents’ trading decisions in line with the market price is determined by a process of trial and error and our experiments have shown scaling parameters in the range 1.5 to 1.7 to be suitable. SENN allows us to place restrictions on the values allowed in the connector via the weight watcher.

Another factor that must be considered in the construction of the model is the number of agents to use in the agents layer. The number of agents used in the model will have consequences on the excess supply or demand in the market. As a start in our model building we use 100 agents in the agents layer of the network. Zimmermann, Grothmann and Neuneier (1997) have done research which indicates that 200 agents is a suitable size to model a foreign exchange market. However, increasing both the size and complexity of the network as such makes it computationally more intensive and increases the time required for learning. For practicality we have chosen to start our model building using 100 agents.

Having provided a brief description of some of the specifics involved in constructing and setting up the neural network model, we now proceed to a description of a very significant part of the model which is learning.

We have already described the significance and importance of learning in a neural network. Various learning rules and paradigms are available for training a neural network. SENN provides different learning algorithms for the training of a neural network. We adapt the following statement of the training problem from the SENN handbook.

Consider a feed forward network with $L$ input nodes and $N$ output nodes. We assume that
there are $T$ data patterns available to train the network. Accordingly, there exists $T$ target vectors $t_1, ..., t_T$ and $T$ input vectors $x_1, ..., x_T$ for which the net computes $T$ output vectors $y_1, ..., y_T$. At this point we are not concerned about the internal structure of the net. An input vector will consist of $N$ components and every output vector will consist of $N$ components. The net contains $p$ weights. The training problem is simply to determine a weight vector that will minimise the error function $E$. We may think of the general procedure to perform this task in terms of the following steps.

1. Randomly choose a starting weight vector $w_0$.

2. Construct a sequence of weight vectors $w_1, w_2, w_3, w_4...$ that will hopefully converge to the position of minimum error.

3. The sequence is obtained by determining two values at each point. Firstly a search direction $d$ and secondly a step length $\eta$, then compute the next iteration point according to the following equation

$$w^{i+1} = w^i + \eta^i d^i$$

The different learning procedures offered by SENN differ in their choice of learning direction and step length.

We proceed to discuss some of these algorithms but first we detail some useful results from mathematical analysis.

We define the error function $E(w)$ as a surface over a two-dimensional weight plane ($p = 2$). From mathematical analysis the following is known:

- The gradient $\nabla E(w^i)$ at the point $E(w^i)$, i.e., the vector of partial derivatives $\nabla E = \left( \frac{\partial E}{\partial w_1}, ..., \frac{\partial E}{\partial w_p} \right)$

points in the direction of steepest ascent of the error function. Similarly, the negative gradient $-\nabla E(w^i)$ points in the direction of steepest descent.

- Any search direction $d^i$ subtending an acute angle with $-E(w^i)$ represents a descent direction.
• The level curves, i.e., the curves of constant $E$ in the $w$-plane, are perpendicular to the local gradient vector at each point. As a result, a learning algorithm will generate a sequence of weights $w^1, w^2, w^3, ...$ with (at worst) non-increasing errors $E(w^1), E(w^2), E(w^3), ...$ when the following conditions are satisfied:

• The search direction $d^t$ makes an acute angle with the negative gradient vector.

• The step length $\eta^t$ is chosen sufficiently small, so as not to miss a local minimum.

Once a sufficiently small neighbourhood of a minimum has been entered, the corresponding algorithm would never leave the neighbourhood. This algorithm will converge in practice to the minimum, as long as the chosen step length is not too small. However, at the start of the training, procedures are chosen intentionally to violate one or both of the above conditions, so that one does not get stuck in a local minimum. If the surface is a paraboloid it will have exactly one minimum (both local and global). This case occurs when the activation function of all clusters is set to identity (i.e., linear mapping).

All SENN's learning procedures make use of the gradient of the error function to determine the search direction in weight space, in addition, some also use higher derivatives or their approximations. To choose a learning procedure we need to make three basic decisions:

1. How many and which patterns are utilised for computation of the gradient?

2. How is the search direction chosen?

3. How is the step length chosen?

We have chosen to use a stochastic pattern selection procedure in which an approximation to the gradient is used to determine the search direction and it is computed from an average over a subset $M$ of all patterns. In this procedure SENN carries out a transition from $w_t$ to $w_t + 1$ every time $|M|$ patterns are read in. $|M|$ denotes the number of elements of $M$. SENN also allows us several methods to choose the subset $M$. We have chosen to use the permute option where during the first adaptive step $x$ patterns are chosen at random. During the second step $x$ patterns are chosen from those that remain and so forth. Each pattern has the same probability of being chosen. Eventually every pattern would have been read in exactly once.
SENN also allows the choice of various procedures for choosing the search direction. We have chosen the VarioEta procedure. This procedure takes the character of the error function as an average of $T$ partial errors more strongly into account. Imagine picking out an arbitrary weight for example the $k$-th. We can interpret the partial derivative

$$\frac{\delta E^t}{\delta w_k}$$

as a force with which the $t$-th pattern pulls on the $k$-th weight. Now if these values strongly differ for $t = 1, ..., T$, i.e., the patterns tend to give inconsistent evaluations of the weight, then our confidence in the information embedded in the partial derivative is reduced. In this case, it is reasonable to perform a relatively small weight adjustment. This idea is implemented by introduction of a weight specific factor

$$\beta_k = \frac{1}{\sum_{t=1}^{T} \left( \frac{\delta E^t}{\delta w_k} - \bar{E} \right)^2} \quad \text{with} \quad E = \frac{1}{T} \sum_{t=1}^{T} \frac{\delta E^t}{\delta w_k}$$

which corresponds (up to the factor $T$) just to the inverse of the scatter of these partial derivatives. The search direction is now determined by multiplying each component of the negative gradient with its beta value, which in vector notation implies that the negative gradient is multiplied by a diagonal matrix.

$$d = - \begin{bmatrix} \beta_1 & 0 & 0 \\ 0 & 0 \\ 0 & 0 & \beta_p \end{bmatrix} \cdot \nabla E$$

The last decision to be made is that of the step length. SENN offers two options for selecting the step length, Fixed step length and LineSearch. We have chosen to use a fixed step length. In this procedure Eta is set by the user to a fixed value usually between 0.01 and 0.1. The problem can be that one does not know a priori what the optimal value is, especially since this value may change the course of the learning process. Very small changes in the error may be due to a value that is too small, while strongly fluctuating or even growing error may be due to a value that is too large. Thus, several different values must be tried. One advantage of the procedure is that it offers a chance to jump out of the neighborhood of a local minimum.

125
and into a hopefully more favourable region.

At the start of the learning process our main objective is not to get stuck in a local minimum and we therefore follow the recommendations of the SENN handbook and choose a stochastic pattern selection procedure, with a VarioEta algorithm to determine search direction and a user defined fixed step length. We have chosen an eta of 0.075 (in keeping with the usual range of 0.01 to 0.1).

We now proceed with a discussion of some of the results extracted from our multi-agent neural network model.

5.2.2 SENN Analysis

Correlation Analysis

Figure 5-4: SENN Correlation Analysis

Figure (5-4) above shows a screen shot of the correlation window where we are able to perform a test of the correlation of particular input series with the target series. Input series with high correlations may have a causal relationship to the target series. Correlation analysis is useful in time series analysis as it allows us to determine times for which the largest influence of the input time series on the target time series is expected. This information is useful as it
allows us to specify time lags in the input file for some time series. Our analysis shows that some inputs like the momentum transformed GBP/ZAR exchange rate tends to show higher correlations with lags and as such model performance could be re-evaluated by specifying time lags for particular inputs. Other input series like both transformations of the dollar gold price tend to show fairly consistent correlations across all time lags and are therefore acceptable to be passed to the agents without a time lag. The correlation analysis therefore allows us to gain some insight into the time lagged relationships between some of the input series and the target series.

Sensitivity Analysis

![Graph of SENN Sensitivity Analysis](image)

If one was using the model for forecasting purposes, sensitivity analysis allows the user to reconstruct point by point the emergence of a forecast and in this way obtain more insight into the structure of the model. The weighting of the individual input series in deriving a forecast is displayed. The user can therefore compare his subjective experience and knowledge of interactions and influences in particular areas with an objective systematic analysis. We are however more concerned with an analysis of the data series and using the neural network
to model the data of a foreign exchange market. Sensitivity actually measures the degree of change at an output node due to changing the value of a particular pattern at a specific input node. The figure above show the sensitivity values at several points plotted as bars. If, over the data space considered, the height of these bars varies only slightly, then over that data space the influence of the relevant data series is fairly linear. Thus, the structural stability of the input series can be determined. We could conduct sensitivity analysis over various data spaces and look for consistency in the stability of the different data series. Additionally, from the sensitivity analysis we are also able to see on which data series different trading strategies place more emphasis, and in this way possibly group the agents in the network into different trading strategies or possibly extract those input series which feature consistently amongst different trading strategies, thereby providing insight into fundamental drivers not previously emphasised.

Figure 5-6: SENN Error Progression Monitor

Figure (5-6) above shows an example of the error window which allows us to monitor the progress of the network error. From this we can observe the influence on the error by changing various network parameters such as the number of agents in the agents layer or alternatively by using different input series. It also gives us an idea of the progression of the error on the
validation and generalization sets as compared to the training set. The progress of the error will give us an indication as to whether we may need to change some of the parameters in our learning algorithm, for example the Eta or fixed step length. In the event that our step length is too large the network error may fluctuate quite substantially. Alternatively if our step length is too small, the error can possibly be trapped in a local minimum on the error surface.

Figure 5-7: SENN Network Evolution Monitor

Figure (5-7) above illustrates the comparison window (network evolution monitor). The comparison window allows us, during training, to carry out a comparison between output and target signals. We are therefore able to obtain evidence for the quality of the model as a representation of reality and thus for the status of the training. What we are also able to observe from this comparison window is that there are periods of the Rand's history where extreme movements have occurred - for example in 2001 an extreme devaluation of the Rand occurred. These extreme movements would represent outliers in the data series and as such must be removed to exclude their influences on the models training.

There are two ways in which the outliers could be removed. We could use a tanh squashing function in our network architecture, which would then remove these outliers. Alternatively, because the extreme period is closer to the end of the data series we have chosen to remove it.
from our analysis and conduct our analysis to the end of the first quarter in 2001.

5.2.3 Discussion of Results

Results for the Initial Data Set

The objective of the exploratory analysis was to investigate the suitability of a multi-agent neural network implementation to model the daily ZAR/USD foreign exchange market. The neural network model that was implemented was a simple 4 layer feed forward model based on explicit market price shifts. The initial time series used was for the period 03/01/1990 to 31/12/2002. Figure 5-8 below shows the graphical representation of the exchange rates movements over this period.

![Graphical Representation of Exchange Rates](image)

**Figure 5-8: Daily ZAR/USD Exchange Rate Time Series**

The first important observation is the unusual spike in the data set in December 2001 when the South African Rand experienced unusual and extreme devaluation against the US Dollar. It is clear that this devaluation is not in line with the long term gradual devaluation of the Rand which according to purchasing power parity may perhaps suggest a value for the exchange rate
closer to between R7.50 and R8.00 to the US Dollar. Many economists at the time believed that the Rand was severely undervalued at intraday levels of close to R14 to the Dollar and a correction was predicted. This correction has since occurred in the market with the Rand testing the R6 to the Dollar mark in late 2003. It would appear that the correction, as with the devaluation has been too severe, and the rand is not yet back on track with its longer term trend as suggested by purchasing power parity. Our opinion is that corrections, which show up as spikes in the historical time series are not optimal for a simple feed forward neural network model. Nevertheless we attempted to evaluate the neural network model using the full time series. Some of the results are presented in Table(5.2) below.

<table>
<thead>
<tr>
<th>ZAR/USD (Jan 1990 to Dec 2002)</th>
<th>Hit Rate</th>
<th>Error Progression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>41.23%</td>
<td>0.014</td>
</tr>
</tbody>
</table>

Table 5.2: SENN Performance Measures on Initial ZAR/USD Data Set

Before proceeding any further it is appropriate to provide a brief explanation of some of the terms that are used in the discussion of our results and concluding remarks. SENN makes provision for built in performance measures which may be used to assess the progression of the neural network during training. Three common performance measures are the accumulated return, the hit rate and the realised potential. SENN does provide additional performance measures, however, we are guided in using the three performance measures above, by Zimmermann, Grothmann and Neuneier, in their research on assessing neural network models. Short descriptions of these three performance measures are given below.

1. **Hit Rate**: The hit rate is a measure of how often the sign of the neural network FX forecast is correctly predicted.

2. **Accumulated Return**: The return accumulated from an agent’s trading strategy.

3. **Realised Potential**: Realised potential is simply the ratio of the model return to the maximum possible return.

4. **Error Progression**: For the purpose of our discussion we define error progression as the best error observed on the generalisation set during training of the neural network.
Our initial analysis for which the results are shown in Table(5.2) above was conducted using 6 inputs in the agents database combined with an agents layer of 200 agents. The neural network was trained and we observe a hit rate of 41.23% on the generalisation set. Additionally a trace of the error progression on the generalisation set showed a best value of 0.0014. Our initial feelings on observing these results was that the hit rate was possibly a little low, having observed higher hit rates in similar neural network implementations of other foreign exchange markets. Based on these results we decided to re-run the model after removing the data spike in December 2001.

As previously mentioned there are predominantly two methods for handling this problem. Firstly a $\tanh$ squashing function may be applied to the data in an attempt to remove any outliers. Alternatively we may shorten the historical time series to exclude the spike and consider a period that is more consistent with the longer term trend of the exchange rate. We opt for the second option for the reasons discussed below.

Firstly, it is our opinion that the extreme devaluation that the Rand experienced in December 2001, is not consistent with the longer term trend that is suggested by purchasing power parity. A similar argument applied to the overcorrection that has recently occurred. Secondly the nature of this multi-agent neural network implementation is in our opinion a relatively simplistic one and may not be well suited to cater for the volatility observed. The analysis is therefore redone on the time series from January 1990 to December 2000.

Revised Results from SENN

The results of the new analysis are presented in Table(5.3) below

<table>
<thead>
<tr>
<th></th>
<th>Hit Rate</th>
<th>Error Progression</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZAR/USD (Jan 1990 to Dec 2000)</td>
<td>50.1%</td>
<td>0.011</td>
</tr>
</tbody>
</table>

Table 5.3: SENN Performance Measures on Revised ZAR/USD Data Set

In comparing the results when using the different time periods we keep constant factors such as the number of inputs available to the agents in the agents database, as well as the number of agents used to model the foreign exchange market. Guided by Zimmerman, Grothmann and Neuneier, we know from their research that these two factors (inputs available/used in the agents database and number of agents used to model the foreign exchange market) are very
important in determining the success of the multi-agent neural network model, when modelling a real world FX market. The results of our investigation of these factors are discussed later. For the slightly shortened data period we observe a hit rate of 50.1% and an error progression of 0.011. The hit rate of 50.1% is a significant improvement on the 41.23% hit rate observed on the longer data period. Similarly the error progression of 0.011 is also an improvement on the 0.014 observed on the longer data set. From these two observations we come to the conclusion that the neural network performance is improved when the extreme movements in the historical time series, caused by the severe devaluation of the rand, are removed. It is possible that the simple 4 layer feed forward network that we have implemented may not be the optimal solution to model a foreign exchange market that experience the type of volatility that is exhibited by the ZAR/USD exchange rate. We however, decided to proceed with our analysis using a 4 layer feed forward neural network, and using a data set that excludes the relatively higher volatility in December 2001.

The next step in our analysis was to attempt to determine:

- whether the number of agents used in the agents layer of the neural network model has an impact on the performance of the model and,

- if possible, the optimal number of agents required to model the ZAR/USD foreign exchange market.

We train our neural network with 100, 200, 300 and 400 agents in the agents layer. The performance of the neural network is evaluated on the basis of hit rate, accumulated return and realised potential. Guided by the improvement in the model performance using a generalisation set that excludes the December 2001 spike, the same data set is used for this analysis. The results of this analysis are presented in the table below.

<table>
<thead>
<tr>
<th>No of Agents</th>
<th>Hit Rate</th>
<th>Realised Potential</th>
<th>Accumulated Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>48.6%</td>
<td>29.8%</td>
<td>6.35</td>
</tr>
<tr>
<td>200</td>
<td>50.1%</td>
<td>25.3%</td>
<td>5.39</td>
</tr>
<tr>
<td>300</td>
<td>49.6%</td>
<td>19.4%</td>
<td>4.13</td>
</tr>
<tr>
<td>400</td>
<td>46.3%</td>
<td>20.2%</td>
<td>4.30</td>
</tr>
</tbody>
</table>

Table 5.4: Comparison of Performance Measures when Number of Agents are varied
From Table(5.4), we observe the highest hit rate of just over 50% (50.1%) for 200 agents. We do however observe a hit rate of close to 50% (49.6%) for 300 agents and hit rates for 100 and 400 agents of 48.6% and 46.3% respectively. From this pattern we may conclude that and agents layer with 200 – 300 agents may be sufficient to model a ZAR/USD foreign exchange market. Examining accumulated return, we observe the highest accumulated return for 100 agents with a return of 6.35%. A very similar return of 5.39% is observed for 200 agents in the agents layer. We see a decline in the accumulated return for 300 and 400 agents with returns of 4.13% and 4.30% respectively. The accumulated return would tend to suggest that possibly between 100 and 200 agents is sufficient to model the ZAR/USD foreign exchange market. The last performance measure examined at was realised potential. By definition the realised potential is derived from the model accumulated return and hence shows a similar pattern as the accumulated return, namely the highest realised potential is achieved for 100 agents, the second highest for 200 agents, with the realised potential falling off for 300 and 400 agents.

Our analysis tends to suggest that a number of agents somewhere between 100 and 200 is sufficient to model the ZAR/USD foreign exchange market. While hit rates remain fairly similar for larger agent sizes (300 and 400) we note that the accumulated return and realised potential tend to fall off for larger number of agents, which is possibly due to larger number of agents employing similar trading strategies and as such reducing the potential for accumulated return and realised potential. We note that Zimmermann, Grothmann and Neuneier, in earlier research have found that 200 agents is sufficient to model the DEM/USD foreign exchange market. From our analysis we reach the conclusion that between 100 and 200 agents would be sufficient to model the ZAR/USD foreign exchange market.

One could also investigate the effects of changing the number of inputs available to the agents in the agents input database. Due to restrictions on the availability of long term time series data we were however constrained to using an agents input database consisting of six inputs for all our analysis. We observe from Zimmermann, Grothmann and Neuneier (1997) that this analysis is best conducted by observing the change in excess demand as the size of the agents data base is varied. We provide a brief description of the results observed by Zimmermann, Grothmann and Neuneier (1997) in their investigations. When a higher average number of inputs is analysed by an agent, the forex market is driven by increasing excess demands, which
implies that if agents are very homogenous in their behaviour then the market tends to be very volatile. Such huge imbalances of supply and demand are inappropriate for adaptation of the underlying dynamics of the foreign exchange market. Research has shown that the best results are obtained by using an agent's database providing on average 9 inputs. On studying the sparse matrix more closely, Zimmermann, Grothmann and Neuneier found that several weights are pushed to zero and the true size of the agents database is on average 7 and not 9 input signals. Most remarkably, an attention span of 7 input signals for human beings is also known in psychology. Based on the results of this research we are fairly comfortable that using 6 input signals in the agents database would not have adversely affected the results of our other analysis.

Our exploratory analysis of neural networks has focused on a multi-agent approach to modelling a foreign exchange market using a simple four layer feed forward network modelled on the explicit market paradigm. Our results and findings therefore focused on analysing and presenting the appropriate factors to model a neural network along these specifications. It is also important to discuss the overall appropriateness of neural networks to modelling multi-agent foreign exchange markets. Thus would be appropriate to compare the results of different types of multi-agent neural network implementations, from a simple three layer MLP, to a four layer network implementing the explicit market paradigm to the more complex error correction neural networks. We have already provided detailed explanations of the explicit market paradigm. Brief overviews of the error correction neural networks and the 3 layer MLP are given next.

5.2.4 Other Neural Network Implementations

Having presented a 4-layer feed forward neural network that attempts to model a foreign exchange market through the explicit market price paradigm, we proceed to provide a brief introduction and discussion on a somewhat more complex structure for a neural network to model a foreign exchange market, i.e., cognitive systems approach of multi-agent foreign exchange market modelling. At this point it is appropriate to make mention of the fact that training a neural network is a computationally intensive task that at times requires many iterations covering many hours if not days before the neural network converges (completes learning). The
sheer computational intensity of these training sessions meant that from a time perspective attempting many different neural network implementations was not feasible.

The principles that we have discussed regarding the dynamics of the market and price formation are still basically the same, namely that the market is basically driven by a superposition of agents decisions, which constitute the average demand or supply. Modelling a foreign exchange market through a cognitive multi-agent systems approach is based on a recurrent neural network architecture which uses an error correction mechanism. A characteristic of the error correction mechanism is that the last model error is offered to the network as an additional input in order to evaluate on its own misspecification. The learning may then interpret the misfit as an external shock which can be used to guide the model dynamics afterward.

**An Introduction to Error Correction Neural Networks**

For discrete time grids, a dynamic system can be described in a very general form by a recurrent state transition equation and an output equation. These equations can be formulated as:

\[ s_t = f(s_{t-1}, u_t) \quad \text{State transition equation} \]

\[ y_t = g(s_t) \quad \text{Output Equation} \]

(5.2)

The state transition is a mapping from the previous internal state of the system \( s_{t-1} \) and the influence of external inputs \( u_t \) to the new state \( s_t \). The output equation gives rise to the observable output \( y_t \). The dynamical system of equation (5.2) can be viewed as a partially observable autoregressive dynamic \( s_t \), which is also driven by external disturbances \( u_t \). Without the external inputs the system is called an autonomous system. The reality however, is that most systems are driven by a superposition of autonomous development and external influences. The task of identifying the dynamical system of equation (5.2) can be stated as the task of finding two functions \( f(.) \) and \( g(.) \) such that an averaged distance measurement (for example equation (5.3)) between the observed data \( y_t^{\text{d}} \) and the models output \( y_t \) is minimal.

\[
\frac{1}{T} \sum_{t=1}^{T} (y_t - y_t^{\text{d}})^2 \rightarrow \min_{f,g}
\]

(5.3)

If we are in the position of having a complete description of all external forces influenc-
ing a deterministic dynamical system the equations (5.2), (5.3) would allow us to identify the temporal relationships by setting up a memory in the form of the state transition equation. Unfortunately, our knowledge about the external forces is typically incomplete or our observations might be noisy. Under such conditions the system identification task using finite data sets often leads to the construction of incorrect causalities. Due to overfitting which we can think of as learning by heart, the generalization properties of such a model are very questionable.

If we are unable to identify the underlying dynamics, because of insufficient input information, we can refer to the last observed model error at \( t - 1 \), which can be interpreted as a warning signal that the model is misleading. Proceeding in this way the last model error is included in the model as an additional input signal. Extending equation (5.2) we obtain equation (5.4)

\[
s_t = f(s_{t-1}, u_t, y_{t-1} - y_{t-1}^d) \\
y_t = g(s_t)
\]

If however, we have a perfect description of the underlying dynamics, the extension of equation (5.4) is no longer required, because the previous observed model error would be zero. In the latter case, one could directly refer to equation (5.2) instead of modelling the dynamic system with equation (5.4). In all other instances the model uses its own error flow as a measurement of unexpected shocks.

A cognitive system has three interdependent basic functionality features namely, perception, internal processing and action. Both perception and action have an overlap with the internal processing. We can think of perception as the observation of the systems environment by the use of all available senses. Perception can be performed consciously or unconsciously where conscious processing can be seen as a comparison between the external world and an autonomous internal picture of the systems environment. Unconscious perception is attributed to the principle of stimulus response.

A further characteristic of perception is that there is no discrimination between input and target information, so a cognitive system treats input and target signals in the same manner.
The Three Layer MLP

The three layer MLP is a three layer neural network. The network consists of one input layer, a hidden layer and an output layer that predicts the market price shifts. This network is provided with the same inputs as the other neural network implementations. The three layer MLP is a neural network implementation that works on the basis of pattern recognition, and is therefore a much simpler implementation of modelling a foreign exchange market.

Studying research done by Zimmermann, Grothmann and Neuneier (1997), we can make the following generalisations about the 3 layer MLP, the 4 layer explicit market paradigm model and the error correction neural network. The three layer MLP, due to it being a pattern recognition model, tends to sometimes display overfitting behaviour due to learning by heart. As a result the model partly loses its generalisation ability and as such is not the most appropriate model to model foreign exchange markets. The 4 layer multi-agent explicit market paradigm model tends to be more dependent on economic and econometric relationships extracted from the input data, while the error correction neural network are able to act and perform in dynamic systems and are not as dependent on external drivers.

We would expect therefore to see an improvement in the results as the implementations move from the simple 3 layer MLP to the error correction neural network. Research and experimentation already conducted verifies that this statement is in actual fact true. The results of our analysis should therefore not be used as a generalisation of the appropriateness or accuracy of neural network implementations in multi-agent modelling of foreign exchange markets. Additionally we must bear in mind that the analysis was conducted on a multi-agent basis but for a single market, i.e., we have only considered the ZAR/USD foreign exchange market. Research has also shown that a multi-agent multi-market analysis can provide further improvements to the results. This approach allows the agents to trade across multiple markets and they are not constrained to trading strategies operating in a single market only. The conclusions provide further explanation of our results as well as suggestions as to what further enhancements may be made to our multi-agent neural network model.
5.2.5 Concluding Remarks

Our findings are therefore, that the ZAR/USD foreign exchange market may be modelled using a feed forward neural network modelled on explicit price dynamics. We recommend using between 100 and 200 agents in the agents layer of the neural network and, using the research of Zimmermann, Grothmann and Neuneier (1997), suggest an average agents database of 7 - 9 input signals. We do however stress that this feed forward neural network modelled on explicit price dynamics may not be the optimal multi-agent neural network model implementation.

In this study we used neural networks to model daily foreign exchange data, so too can neural networks be used to model intraday data. However, due to time constraints we did not apply the neural networks to model intraday data, as only daily data was initially available and intraday data became available at a much later stage.

Neural networks as a class of model, for modelling the dynamics of a foreign exchange market have vast potential, the bounds of which have not been tested by the relatively simplified neural network model implemented in this exploratory analysis. Further analysis using more complex neural network implementations such as error correction neural networks, as well as adding the extra dimension of multi-market trading, should show that a multi-agent neural network approach to modelling foreign exchange markets is appropriate.

We now proceed to examine the suitability of the Kalman Filter for analysing the dynamics of the ZAR/USD exchange rate.
Chapter 6

THE KALMAN FILTER IN FINANCE

The aim of this chapter is to utilise the Kalman filter to model the DEM/USD and ZAR/USD returns time series, as stated in the research objectives.

6.1 Introduction to State-Space Models

When considering system analysis or controller design, why go beyond deterministic theories and propose stochastic system models? Deterministic system and control theories do not provide a totally sufficient means of performing analysis and design. Maybeck (1979) discusses the three shortcomings of deterministic theories. *No mathematical system model is perfect.* It is often the case that the “laws” of Newtonian physics provide adequate system structures, but the various parameters within that structure are not determined absolutely, which results in many sources of uncertainty in a mathematical model of a system. Secondly, deterministic systems are driven not only by control inputs, but also by *disturbances which can neither be controlled nor modelled deterministically.* And finally, *sensors do not provide perfect and complete data* about a system. Sensors do not provide exact readings of desired quantities, but rather introduce their own system dynamics and distortions. These devices are usually noise corrupted.

A Kalman filter is an optimal recursive data processing algorithm. The Kalman filter
processes all available measurements to estimate the current value of the desired variables in such a way that the error is minimised statistically, with the use of knowledge of the system and measurement device dynamics; the statistical description of system noises, measurement errors, and uncertainty in the dynamic models; and any available information about the initial conditions of the variables of interest. It does not require all previous data to be stored and reprocessed every time a new measurement is taken.

The Kalman filter is named after Rudolph E. Kalman, who in 1960 published his paper describing a recursive solution to the discrete-data linear filtering problem. The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance. The Kalman filter has been extensively researched and applied to the area of autonomous navigation.

The state space form is the key to handling structural time series models, with the state of the system representing the various unobserved components, for example, trends. Once in state space form, the Kalman filter provides the means of updating the state as new observations become available. For the purpose of prediction, these components are extrapolated into the future. Various smoothing techniques can be used to obtain the best estimate of the state at any point within the sample. The prediction and smoothing can only be applied after the parameters governing the stochastic dynamics of the state variables have been estimated. The estimation of these hyperparameters is itself based on the Kalman filter, since the likelihood function can be expressed in terms of one-step ahead prediction errors and the prediction errors are a by-product of the filter.

The Kalman filter is useful in some engineering applications because of on-line estimation. The current value of the state vector is of prime interest and the Kalman filter enables the estimate of the state vector to be continually updated as new observations become available. Sometimes the state vector does not have an economic interpretation but, if it does, it is more appropriate to estimate its value at a specific point in time using all rather than some information in the sample. The Kalman filter may then be applied to the economic and econometric areas of prediction and smoothing. If the disturbances and initial state vector are normally distributed, the Kalman filter enables the likelihood function to be calculated via the prediction error decomposition, which results in the estimation of unknown parameters in the model.
The modelling of time series in state space form has advantages over other techniques both in interpretability and estimation. The Kalman filter lies at the heart of state space analysis and provides the basis for likelihood estimation. The state space form is advantageous in that it can handle missing observations and temporal aggregation. In addition it is possible to modify structural time series models into continuous time. As a result the models become independent of the time interval between observations, enabling the Kalman filter to be applied to irregularly spaced observation samples.

The main difficulty in empirically testing particular time series models is that frequently the underlying variables are not directly observable. The state space form is the appropriate procedure to deal with situations in which the state variables are not observable, but known to be generated by a Markov process. The Kalman filter generates estimates for the state variables and also facilitates the calculation of the likelihood of observing a particular data series given a specified set of model parameters. Maximum likelihood estimation can then be used to determine the optimal set of parameters.

6.1.1 The Discrete Kalman Filter Algorithm

The Kalman filter estimates a process by using a form of feedback control, i.e., the filter estimates the process state at some time and then obtains feedback in the form of noisy measurements. The Kalman filter equations can be grouped into: time update equations and measurement update equations. The time update equations are responsible for projecting forward in time the current state and error covariance estimates to obtain the a priori estimates for the next step. Then the measurement update equations incorporate a new measurement into the a priori estimate to obtain an improved a posteriori estimate. Thus the time update equations can be thought of as predictor equations, whilst the measurement equations can be thought of as corrector equations. After every time and measurement update pair, this process is repeated and the previous a posteriori estimates used to project or predict the next a priori estimates. This recursive nature of the Kalman filter is what makes it so appealing to implement. Practical implementations are much more feasible than, for example, the implementation of a Wiener filter (Brown and Hwang 1996) which is designed to operate on all of the data directly for each estimate. Instead, the Kalman filter recursively conditions the
current estimate on all of the past measurements.

6.2 The State-Space Representation of a Dynamic System

The idea behind a state-space representation of a complicated linear system is to capture the
dynamics of an observed \((n \times 1)\) vector \(y_t\) in terms of a possibly unobserved \((r \times 1)\) vector \(\xi_t\) known as the state vector for the system. The state-space representation of the dynamics of
the state vector is given by the following equation:

\[
\xi_{t+1} = F\xi_t + v_{t+1}
\]  

(6.1)

Here \(F\) is a matrix of dimension \((r \times r)\), and the \((r \times 1)\) vector \(v_t\) is taken to be independent and identically distributed where

\[E(v_tv'_t) = \begin{cases} Q & \text{for } t = \tau \\ 0 & \text{otherwise} \end{cases}\]

and \(Q\) is a \((r \times r)\) matrix. Future values of the state vector depend on \((\xi_t, \xi_{t-1}, \ldots)\) only through
the current value \(\xi_t\). If the eigenvalues of \(F\) all lie inside the unit circle, then the system is
stable. The observed variables are related to the state vector through the system’s observation equation:

\[
y_t = A'x_t + H'\xi_t + w_t
\]  

(6.2)

Here \(y_t\) is an \((n \times 1)\) vector of variables observed at time \(t\), \(H'\) is an \((n \times r)\) matrix of coefficients, and the \((n \times 1)\) vector \(w_t\) is independently and identically distributed with

\[E(w_tw'_t) = \begin{cases} R & \text{for } t = \tau \\ 0 & \text{otherwise} \end{cases}\]

and \(R\) is a \((r \times r)\) matrix. \(x_t\) is a \((k \times 1)\) vector of observed exogenous or predetermined
variables, which enter the state equation through the \((n \times k)\) matrix of coefficients \(A'\). The
exogenous or predetermined \(x_t\) implies that \(x_t\) provides no information about \(\xi_{t+s}\) or \(w_{t+s}\)
for \( s = 0, 1, 2, \ldots \) beyond that contained in \( y_{t-1}, y_{t-2}, \ldots, y_1 \). The disturbances \( v_t \) and \( w_t \) are assumed to be uncorrelated at all lags:

\[
E(v_tw'_t) = 0, \text{ for all } t \text{ and } \tau
\]

The state and observation equation constitute a linear state space representation for the dynamic behaviour of \( y \). This concept can be further adapted to allow for time-varying coefficient matrices, non-normal disturbances and nonlinear dynamics, which are discussed later in this chapter.

6.2.1 Derivation of the Kalman Filter

The matrices \( A, H \) and \( R \) in the observation (measurement) equation and the matrices \( F \) and \( Q \) in the state (transition) equation is referred to as the system matrices. For now it will be assumed that they are non-stochastic, i.e., although they can change with time they do so in a way which is predetermined. Allowing these system matrices to depend on past observations is explored later in this chapter. If the system matrices do not change over time, the model is said to be time-invariant or time-homogenous.

The derivation below is based on the assumption that the disturbances and initial state vector are normally distributed. After deriving the Kalman filter, it is shown that the mean of the conditional distribution of \( \xi_{t+1} \) is an optimal estimator of \( \xi_{t+1} \) as it minimises the mean square error (MSE). If the normality assumption is dropped, the Kalman filter is still an optimal estimator because it minimises the MSE within the class of all linear estimators.

We assume that the values of all the elements of \( F, Q, A, H \) and \( R \) are known with certainty. The Kalman filter can be described as an algorithm for calculating linear least squares forecasts of the state vector on the basis of data observed through date \( t \),

\[
\hat{\xi}_{t+1|t} = \hat{E}(\xi_{t+1}|\zeta_t)
\]

where

\[
\zeta_t = (y'_t, y'_{t-1}, \ldots, y'_1, x'_t, x'_{t-1}, \ldots, x'_1)'
\]

144
and \( \hat{E}(\xi_{t+1}|\xi_t) \) denotes the linear projection of \( \xi_{t+1} \) on \( \xi_t \) and a constant. The Kalman filter recursively calculates these forecasts, generating \( \hat{\xi}_{1|0}, \hat{\xi}_{2|1}, \ldots, \hat{\xi}_{T|T-1} \) in succession. A mean squared error (MSE) \((r \times r)\) matrix is associated with each of these forecasts:

\[
P_{t+1|t} = E[(\xi_{t+1} - \hat{\xi}_{t+1|t})(\xi_{t+1} - \hat{\xi}_{t+1|t})']
\]

**Starting the Recursion**

To start the iteration it is assumed that the initial value of the state vector \( \xi_1 \) is drawn from a normal distribution. The recursion begins with \( \hat{\xi}_{1|0} \), the unconditional mean of \( \xi_1 \),

\[
\hat{\xi}_{1|0} = E(\xi_1)
\]

with associated MSE

\[
P_{1|0} = E[(\xi_1 - E(\xi_1))(\xi_1 - E(\xi_1))']
\]

If the eigenvalues of \( F \) are all inside the unit circle, then the vector process for \( \xi_t \) in (6.1) is covariance-stationary. By taking expectations of both sides of (6.1), the unconditional mean of \( \xi_t \) can be found:

\[
E(\xi_{t+1}) = F \cdot E(\xi_t)
\]

but \( \xi_t \) is mean-stationary, thus

\[
(I_r - F) \cdot E(\xi_t) = 0
\]

This equation has the unique solution \( E(\xi_t) = 0 \) (Since unity is not an eigenvalue of \( F \), the matrix \( (I_r - F) \) is nonsingular). Then \( \hat{\xi}_{1|0} = 0 \). The unconditional variance can similarly be found by postmultiplying (6.1) by its transpose and taking expectations:

\[
E(\xi_{t+1}\xi'_t) = E[(F\xi_t + v_{t+1})(\xi'_t F' + v'_{t+1})] = F \cdot E(\xi_t \xi'_t) \cdot F' + E(v_{t+1}v'_{t+1})
\]

If \( \xi_t \) is stationary, then \( E(\xi_{t+1}\xi'_t) = E(\xi_t \xi'_t) = P_{1|0} \) and the equation above is reduced to

145
\[ P_{1|0} = FP_{1|0}F' + Q \]

Then apply the vec operator to the above equation and recall [e.g. Magnus and Neudecker (1988, pg 30)] that \( \text{vec}(ABC) = (C' \otimes A) \text{vec}(B) \) produces \( \text{vec}(P_{1|0}) = (F \otimes F) \cdot \text{vec}(P_{1|0}) + \text{vec}(Q) \). This unconditional variance is calculated from \( \text{vec}(P_{1|0}) = [I_{r^2} - (F \otimes F)]^{-1} \cdot \text{vec}(Q) \). \( I_{r^2} \) is the \( (r^2 \times r^2) \) identity matrix, \( \otimes \) denotes the Kronecker product and \( \text{vec}(P_{1|0}) \) is the \( (r^2 \times 1) \) vector formed by stacking columns of \( P_{1|0} \), one on top of the other, ordered from left to right.

If the system is nonstationary or time-variant then \( \hat{\xi}_{1|0} \) could represent a guess as to the value of \( \xi_t \) based on prior information and \( P_{1|0} \) measures the uncertainty associated with this guess. Larger values for the diagonal elements of \( P_{1|0} \) are a result of greater uncertainty of \( \xi_t \).

**Forecasting \( y_t \)**

Given the starting values \( \hat{\xi}_{1|0} \) and \( P_{1|0} \), we then calculate \( \hat{\xi}_{2|1} \) and \( P_{2|1} \), i.e., given \( \hat{\xi}_{t|t-1} \) and \( P_{t|t-1} \) we need to evaluate \( \hat{\xi}_{t+1|t} \) and \( P_{t+1|t} \).

Since we assume that \( x_t \) contains no information about \( \xi_t \) beyond that contained in \( \zeta_t \),

\[ \hat{E}(\xi_t|x_t, \zeta_{t-1}) = \hat{E}(\xi_t|\zeta_{t-1}) = \hat{\xi}_{t|t-1} \]

Then, forecasting the value of \( y_t \):

\[ \hat{y}_{t|t-1} = \hat{E}(y_t|x_t, \zeta_{t-1}) \]

From equation (6.2)

\[ \hat{E}(y_t|x_t, \xi_t) = A'x_t + H'\hat{\xi}_t \]

And using the law of iterated projections,

\[ \hat{y}_{t|t-1} = A'x_t + H' \cdot \hat{E}(\hat{\xi}_t|x_t, \zeta_{t-1}) = A'x_t + H'\hat{\xi}_{t|t-1} \]

From equation (6.2) the error of this forecast is

146
\[ y_t - \hat{y}_{t|t-1} = A'x_t + H'\xi_t + w_t - A'x_t - H'\hat{\xi}_{t|t-1} \]
\[ = H'(\xi_t - \hat{\xi}_{t|t-1}) + w_t \] (6.4)

with MSE

\[ E[(y_t - \hat{y}_{t|t-1})(y_t - \hat{y}_{t|t-1})'] = E[H'(\xi_t - \hat{\xi}_{t|t-1})(\xi_t - \hat{\xi}_{t|t-1})']H + E[w_tw'_t] \]
\[ = H'P_{t|t-1}H + R \]

Similarly, the conditional variance between equation (6.4) and the error in forecasting the state vector is

\[ E\{y_t - E(y_t|x_t, \zeta_{t-1})|\zeta_t = E(\xi_t|x_t, \zeta_{t-1})|x_t, \zeta_{t-1}\} \]
\[ = H' \cdot E\{[\xi_t - \hat{\xi}_{t|t-1}][\xi_t - \hat{\xi}_{t|t-1}]'|x_t, \zeta_{t-1}\} \]
\[ = H'P_{t|t-1} \]

Thus the distribution of the vector \((y_t', \xi_t')'\) conditional on \(x_t\) and \(\zeta_{t-1}\) is

\[
\begin{bmatrix}
  y_t | x_t, \zeta_{t-1} \\
  \xi_t | x_t, \zeta_{t-1}
\end{bmatrix}
\sim N\left(\begin{bmatrix}
  A'x_t + H'\hat{\xi}_{t|t-1} \\
  \hat{\xi}_{t|t-1}
\end{bmatrix}, \begin{bmatrix}
  H'P_{t|t-1}H + R & H'P_{t|t-1} \\
  P_{t|t-1}H & P_{t|t-1}
\end{bmatrix}\right)
\]

It then follows that \(\xi_t|\zeta_t = \xi_t|x_t, y_t, \zeta_{t-1}\) is distributed \(N(\hat{\xi}_{t|t|P_{t|t}})\) where

\[ \hat{\xi}_{t|t} = \hat{\xi}_{t|t-1} + P_{t|t-1}H(H'P_{t|t-1}H + R)^{-1}(y_t - A'x_t - H'\hat{\xi}_{t|t-1}) \] (6.5)

\[ P_{t|t} = P_{t|t-1} - P_{t|t-1}H(H'P_{t|t-1}H + R)^{-1}H'P_{t|t-1} \] (6.6)

The final step is to calculate a forecast of \(\xi_{t+1}\) conditional on \(\zeta_t\). From equation (6.1) it can
be seen that
\[ f_{t+1|t} \sim N(\hat{f}_{t+1|t}, P_{t+1|t}) \]
where
\[ \hat{f}_{t+1|t} = \hat{f}_{t|t} \]  
(6.7)
\[ P_{t+1|t} = FP_{t|t}F' + Q \]  
(6.8)
Substituting (6.5) into (6.7) and (6.6) into (6.8),
\[ \hat{f}_{t+1|t} = \hat{f}_{t|t} + Fp_{t|t}H'H(f_{t|t} + R)^{-1}(y_t - A'x_t - H'\hat{f}_{t|t}) \]  
(6.9)
\[ P_{t+1|t} = FP_{t|t}F' - FP_{t|t}H'H(f_{t|t} + R)^{-1}H'P_{t|t}F' + Q \]
The coefficient matrix in equation (6.9) is known as the gain matrix and is denoted \( K_t \):
\[ K_t = FP_{t|t}H'H(f_{t|t} + R)^{-1} \]
In summary, the Kalman filter is an algorithm for calculating the sequences \( \{\hat{f}_{t+1|t}\}_{t=1}^{T} \) and \( \{P_{t+1|t}\}_{t=1}^{T} \). \( \hat{f}_{t+1|t} \) denotes the optimal forecast of \( f_{t+1} \) based on observation of \( (y_t, y_{t-1}, \ldots, y_1, x_t, x_{t-1}, \ldots) \), and \( P_{t+1|t} \) denotes the mean square error of the forecast. Since the sequence \( \{P_{t+1|t}\}_{t=1}^{T} \) is not a function of the data, it can be evaluated without calculating the forecasts \( \{\hat{f}_{t+1|t}\}_{t=1}^{T} \).
Then the conditional expectation of the squared forecast error is the same as its unconditional expectation, \( P_{t+1|t} = E[(f_{t+1} - \hat{f}_{t+1|t})(f_{t+1} - \hat{f}_{t+1|t})'] = E[(f_{t+1} - \hat{f}_{t+1|t})(f_{t+1} - \hat{f}_{t+1|t})'] \). This results from the assumption of normal distributions with constant variances for \( v_t \) and \( w_t \).

6.2.2 Forecasting with the Kalman filter

The forecast of \( y_t \) calculated in equation (6.3) is an exact finite-sample forecast of \( y_t \) on the basis of \( x_t \) and \( \zeta_{t-1} \equiv (y_{t-1}', y_{t-2}', \ldots, y_1', x_{t-1}', x_{t-2}', \ldots, x_1')' \). If \( x_t \) is deterministic, the Kalman filter can be used to calculate exact finite-sample \( s \)-period-ahead forecasts.

The state equation (6.1) can be solved by recursive substitution to yield
$$\xi_{t+s} = F^s \xi_t + F^{s-1} v_{t+1} + F^{s-2} v_{t+2} + ... + F^1 v_{t+s-1} + v_{t+s} \quad \text{for } s = 1, 2, ...$$  \hfill (6.10)

The projection of $\xi_{t+s}$ on $\xi_t$ and $\xi_{t+s}$ is given by

$$\hat{E}(\xi_{t+s}|\xi_t, \xi_{t+s}) = F^s \xi_t$$  \hfill (6.11)

From the law of iterated projections,

$$\hat{\xi}_{t+s|t} \equiv \hat{E}(\xi_{t+s}, \xi_t) = F^s \hat{\xi}_{t|t}$$  \hfill (6.12)

Thus, from equation (6.10) the $s$-period-ahead forecast error for the state vector is

$$\xi_{t+s} - \hat{\xi}_{t+s|t} = F^s (\xi_t - \hat{\xi}_{t|t}) + F^{s-1} v_{t+1} + F^{s-2} v_{t+2} + ... + F^1 v_{t+s-1} + v_{t+s} \quad \text{for } s = 1, 2, ...$$

with MSE

$$P_{t+s|t} = F^s P_{t|t} (F')^s - F^{s-1} Q (F')^{s-1} + F^{s-2} Q (F')^{s-2} + ... + F Q F' + Q$$

Forecasting the observed vector $y_{t+s}$

$$\hat{y}_{t+s|t} \equiv \hat{E}(y_{t+s}, \xi_t) = A' x_{t+s} + H' \hat{\xi}_{t+s|t}$$

The error of this forecast is

$$y_{t+s} - \hat{y}_{t+s|t} = (A' x_{t+s} + H' \hat{\xi}_{t+s|t} + w_{t+s}) - (A' x_{t+s} + H' \hat{\xi}_{t+s|t}) \quad \hfill (6.13)$$

$$= H'(\xi_{t+s} - \hat{\xi}_{t+s|t}) + w_{t+s}$$

with MSE
\[ E[(y_{t+s} - \hat{y}_{t+s}|t})(y_{t+s} - \hat{y}_{t+s}|t)'] = H'P_{t+s|t}H + R \]

6.2.3 Statistical Inference about Unknown Parameters using the Kalman filter

**Maximum Likelihood Estimation**

Maximum Likelihood (ML) theory is based on a scenario in which the \( T \) sets of observations, \( y_1, ..., y_T \) are independently and identically distributed. The joint density function is given by

\[ L(y; \psi) = \prod_{t=1}^{T} p(y_t) \]

where \( p(y_t) \) is the (joint) probability density function (pdf) of the \( t \)-th set of observations. \( L(y; \psi) \) then becomes the likelihood function, after the observations have been made, and the ML estimator is found by maximising this function with respect to \( \psi \). This function is not applicable to time series models since the observations are not independent. Thus the conditional probability function is used to write the joint density function as

\[ L(y; \psi) = \prod_{t=1}^{T} p(y_t|Y_{t-1}) \]

where \( p(y_t|Y_{t-1}) \) denotes the distribution of \( y_t \) conditional on the information set as time \( t - 1 \), that is \( Y_{t-1} = \{y_{t-1}, y_{t-2}, ..., y_1\} \).

The forecasts \( \hat{\xi}_{t|t-1} \) and \( \hat{\eta}_{t|t-1} \) are optimal within the set of forecasts that are linear in \((x_t, \zeta_{t-1})\), where \( \zeta_{t-1} = (y_{t-1}', y_{t-2}', ..., y_1', x_{t-1}', x_{t-2}', ..., x_1')' \). When the initial state \( \xi_1 \) and the innovations \( \{w_t, v_t\}_{t=1}^{T} \) are multivariate Gaussian, then the forecasts \( \hat{\xi}_{t|t-1} \) and \( \hat{\eta}_{t|t-1} \) calculated by the Kalman filter are optimal among any functions of \((x_t, \zeta_{t-1})\). In addition, if \( \xi_1 \) and \( \{w_t, v_t\}_{t=1}^{T} \) are Gaussian, then the distribution of \( y_t \) conditional on \((x_t, \zeta_{t-1})\) is Gaussian with mean given by equation (6.10) and variance given by equation (6.11):

\[ y_t|x_t, \zeta_{t-1} \sim N \left( (A'x_t + H'\hat{\xi}_{t|t-1}), (H'P_{t|t-1}H + R) \right) \]

That is,
\[ f_{y_t|x_t, \xi_{t-1}}(y_t|x_t, \xi_{t-1}) = (2\pi)^{-(n/2)}|H'P_{t|t-1}H + R|^{-1/2} \]
\[ \times \exp\left\{ -\frac{1}{2}(y_t - A'x_t - H'\hat{\xi}_{t|t-1})'(H'P_{t|t-1}H + R)^{-1}(y_t - A'x_t - H'\hat{\xi}_{t|t-1}) \right\} \text{ for } t = 1, 2, ..., T \]

The sample log likelihood

\[ \sum_{t=1}^{T} \log f_{y_t|x_t, \xi_{t-1}}(y_t|x_t, \xi_{t-1}) \]  

(6.15)

can then be maximised numerically with respect to the unknown parameters in the matrices \( F, Q, A, H \) and \( R \).

The derivatives of the log likelihood required for numerical search procedures for maximising equation (6.15) are calculated numerically or analytically. To characterise the analytical derivatives of equation (6.15), collect the unknown parameters to be estimated in a vector \( \theta \), and write \( F(\theta), Q(\theta), A(\theta), H(\theta) \) and \( R(\theta) \). Implicitly, \( \hat{\xi}_{t|t-1}(\theta) \) and \( P_{t|t-1}(\theta) \) will be functions of \( \theta \) too. The derivative of the log of equation (6.14) with respect to the \( i \)-th element of \( \theta \) involves \( \partial \hat{\xi}_{t|t-1}(\theta)/\partial \theta_i \) and \( \partial P_{t|t-1}(\theta)/\partial \theta_i \).

Identification

The maximum likelihood procedure presupposes that the model is identified, that is, it assumes that a change in any of the parameters would imply a different probability distribution for \( \{y_t\}_{t=1}^{\infty} \).

Asymptotic properties of maximum likelihood estimates

IF \( \hat{\theta} \) denotes the maximum likelihood estimator, then under suitable conditions, \( \hat{\theta} \) is consistent and asymptotically normal. These conditions include:

- \( \theta \) is identifiable;
- eigenvalues of \( F \) are inside the unit circle;
• the exogenous variable $x_t$ behaves asymptotically as a full rank linearly nondeterministic covariance-stationary process;

• the true value of $\theta$ does not fall on the boundary of the allowable parameter space.

Pagan (1980, Theorem 4) and Ghosh (1989) demonstrated that for particular examples of state-space models

$$\sqrt{T} \theta_{2D,T}^{1/2} (\hat{\theta} - \theta_0) \sim N(0, I)$$

where $\theta_{2D,T}$ is the information matrix for a sample of size $T$ as calculated from second derivatives of the log likelihood function:

$$\theta_{2D,T} = -\frac{1}{T} E \left[ \sum_{t=1}^{T} \frac{\partial^2 \log f(y_t|x_{t-1}, x_t; \theta)}{\partial \theta \partial \theta'} \bigg| \theta = \theta_0 \right]$$

A common practice is to assume that the limit of $\theta_{2D,T}$ as $T \to \infty$ is the same as the plim of

$$\hat{\theta} = -\frac{1}{T} \sum_{t=1}^{T} \frac{\partial^2 \log f(y_t|x_{t-1}, x_t; \theta)}{\partial \theta \partial \theta'} \bigg|_{\theta = \hat{\theta}}$$

which can be calculated analytically or numerically by differentiating equation (6.15). Reported standard errors for $\hat{\theta}$ are then the square roots of diagonal elements of $(1/T)(\hat{\theta})^{-1}$.

Even if the disturbances $v_t$ and $w_t$ are non-Gaussian, the Kalman filter can still be used to calculate the linear projection of $y_{t+s}$ on past observations. It is thus of interest to consider what happens if we use as an estimate of $\theta$ the value that maximises equation (5.2), even though the true distribution is not normal. Under certain conditions such quasi-maximum likelihood estimates give consistent and asymptotically normal estimates of the true value of $\theta$, with

$$\sqrt{T} (\hat{\theta} - \theta_0) \sim N(0, [\theta_{2D} \theta_{OP}^{-1} \theta_{2D}]^{-1})$$

where $\theta_{2D}$ is the plim when evaluated at the true value $\theta_0$ and $\theta_{OP}^{-1}$ is the limit of

$$(1/T) \sum_{t=1}^{T} [s_t(\theta_0)] [s_t(\theta_0)]'$$

152
where

\[ s_t(\theta_0) \equiv \left[ \frac{\partial \log f(y_t|\zeta_{t-1}, x_t; \theta)}{\partial \theta} \right]_{\theta=\theta_0} \]

### 6.2.4 Smoothing

Filtering aims to find the expected value of the state vector, \( \xi_t \), conditional on the information available at time \( t \). The aim of smoothing is to take account of the information made available after time \( t \). The mean of the distribution of \( \xi_t \), conditional on the sample, is known as the smoothed estimate. This smoothed estimate of \( \xi_t \) may be denoted as \( \hat{\xi}_{t|T} \equiv \hat{E}(\xi_t|\zeta_T) \). The corresponding estimator is called a smoother. The smoother is based on more information than the filtered estimator, and will therefore have a MSE smaller than that of the filtered estimator. The MSE of the smoothed estimate is denoted \( P_{t|T} \equiv E[(\xi_t - \hat{\xi}_{t|T})][(\xi_t - \hat{\xi}_{t|T})'] \).

There are basically three smoothing algorithms in a linear model. Fixed-point smoothing computes smoothed estimates of the state vector at some fixed point in time. Thus it gives \( \hat{\xi}_{\tau|t} \) for particular values of \( \tau \) at all time periods \( t > \tau \). Fixed-lag smoothing computes estimates for a fixed delay, that is \( \hat{\xi}_{t-j|t} \) for \( j = 1, \ldots, M \) where \( M \) is a maximum lag. Both fixed-point smoothing and fixed-lag smoothing can be applied in an on-line situation. Fixed-interval smoothing is concerned with computing the full set of smoothed estimates for a fixed span of data. Since fixed-interval smoothing is an off-line technique, which yields \( \hat{\xi}_{t|T}, t = 1, \ldots, T \), it tends to be the most widely used algorithm for economic and social data. Fixed-point smoothing can be used in an off-line situation where the state vector only needs to be estimated at a limited number of points in time.

These three recursive algorithms are closely linked to the Kalman filter. The fixed-point algorithm runs in parallel with the Kalman filter, while the fixed-interval algorithm is a backward recursion which starts at time \( T \), and produces the smoothed estimates in the order \( T, \ldots, 1 \).

In some applications the value of the state vector is of interest in its own right. In such cases it is desirable to use information through the end of the sample (date \( T \)) to help improve the inference about the historical value that the state vector took on at any particular date \( t \) in the middle of the sample. Such as inference is known as a smoothed estimate.

The smoothed estimates are calculated in the following manner. The data is run through
the Kalman filter, storing the sequences \( \{P_{t|t}\}_{t=1}^{T} \) and \( \{P_{t|t-1}\}_{t=1}^{T} \) calculated from equations (6.6) and (6.8), and storing the sequences \( \{\xi_{t|t}\}_{t=1}^{T} \) and \( \{\xi_{t|t-1}\}_{t=1}^{T} \) calculated from equations (6.5) and (6.7). The final value for \( \{\xi_{t|t}\}_{t=1}^{T} \) then gives the smoothed estimate for the last date in the sample, \( \hat{\xi}_{T|T} \), and \( P_{T|T} \) is the MSE. The sequence of smoothed estimates \( \{\hat{\xi}_{t|T}\}_{t=1}^{T} \) is then calculated in reverse order by iterating on

\[
\hat{\xi}_{t|T} = \hat{\xi}_{t|t} + J_t(\hat{\xi}_{t+1|T} - \hat{\xi}_{t+1|t}) \quad \text{for} \quad t = T - 1, T - 2, \ldots, 1
\]

where \( J_t = P_{t|t} F_{t+1}^{-1} \). Similarly the corresponding MSE are found by iterating on

\[
P_{t|T} = P_{t|t} + J_t(P_{t+1|T} - P_{t+1|t})J_t^t \quad \text{for} \quad t = T - 1, T - 2, \ldots, 1
\]
in reverse order.

Confidence Intervals for Smoothed Estimates and Forecasts

Let \( \hat{\xi}_{T|T}(\theta_0) \) denote the optimal inference about \( \xi_T \) conditional on observation of all data through date \( T \), assuming that \( (\theta_0) \) is known. For \( \tau \leq T \), \( \hat{\xi}_{T|T}(\theta_0) \) is the smoothed inference with \( P_{T|T}(\theta_0) \) denoting the mean squared error of this inference, while for \( \tau > T \), \( \hat{\xi}_{T|T}(\theta_0) \) is the forecast with \( P_{T|T}(\theta_0) \) as the mean squared error.

If the true value of \( \theta \) is unknown, the optimal inference is approximated by \( \hat{\xi}_{T|T}(\hat{\theta}) \) for \( \hat{\theta} \) the maximum likelihood estimate. Conditional on having observed all the data \( \zeta_t \), the posterior distribution might be approximated by

\[
\theta|\zeta_t \sim N(\hat{\theta}, (1/T)(\hat{\theta})^{-1})
\]  \hspace{1cm} (6.16)

Hamilton (1994) showed that

\[
E\{[\xi_T - \hat{\xi}_{T|T}(\hat{\theta})][\xi_T - \hat{\xi}_{T|T}(\hat{\theta})]'|\zeta_t\} = \text{E}_{\theta|\zeta_t}\{[\xi_T - \hat{\xi}_{T|T}(\theta)][\xi_T - \hat{\xi}_{T|T}(\theta)]'|\zeta_t\} \\
\quad + \text{E}_{\theta|\zeta_t}\{[\hat{\xi}_{T|T}(\theta) - \hat{\xi}_{T|T}(\hat{\theta})][\hat{\xi}_{T|T}(\theta) - \hat{\xi}_{T|T}(\hat{\theta})]'|\zeta_t\}
\]

154
where $E_{\theta|\zeta_t}(\cdot)$ denotes the expectation of $(\cdot)$ with respect to the distribution in equation (6.16). Thus the mean squared error of an inference based on estimated parameters is the sum of two terms. The first term can be written as $E_{\theta|\zeta_t}\{P_{\tau|T}(\theta)\}$, and may be described as "filter uncertainty". A good method of calculation would be to generate, for example, 10,000 Monte Carlo draws of $\theta$ from the distribution in equation (6.16). Then run through the Kalman filter iterations implied by each draw, and calculate the average value of $P_{\tau|T}(\theta)$ across draws. The second term, may be described as "parameter uncertainty", and could be estimated from the outer product of $[\hat{\xi}_{t|T}(\theta_i) - \hat{\xi}_{t|T}(\hat{\theta})]$ with itself for the $i$-th Monte Carlo draw, and again averaging across Monte Carlo realisations.

### 6.2.5 Time-Varying Coefficient Models

Until now we have been assuming that the matrices $F$, $Q$, $A$, $H$, and $R$ were all constant. The Kalman filter can also be generalised for state space models in which the values of these matrices depend on exogenous or lagged dependent variables of $x_t$.

\[
\begin{align*}
\xi_{t+1} &= F(x_t)\xi_t + v_{t+1} \\
E(v_{t+1}v_{t+1}'|\zeta_t) &= Q(x_t) \\
y_t &= a(x_t) + [H(x_t)]'\xi_t + w_t \\
E(w_tw_t'|\zeta_{t-1}) &= R(x_t)
\end{align*}
\] (6.17)

Here $F(\cdot)$, $Q(\cdot)$, $H(\cdot)$ and $R(\cdot)$ denote matrix-valued functions of $x_t$ and $a(\cdot)$ is an $(n \times 1)$ vector-valued function of $x_t$. It is assumed that $x_t$ provides no information about $\xi_t$ or $w_t$ for any $t$ beyond that contained in $\zeta_{t-1}$ (apart from the possible heteroskedasticity implied in equation (6.18)).

The unconditional distributions of $\xi_t$ and $y_t$ are no longer normal with $x_t$ stochastic even if $v_t$ and $w_t$ are normal. However, the system is conditionally normal. If the distribution of $\xi_t$ conditional on $\zeta_{t-1}$ is $N(\hat{\xi}_{t|\zeta_{t-1}}, P_{t|\zeta_{t-1}})$, then $\xi_t$ conditional on $x_t$ and $\zeta_{t-1}$ has the same distribution. Conditional on $x_t$ all of the matrices can be treated as deterministic.

Therefore the derivation of the Kalman filter is essentially as before,
\[ \hat{\xi}_{t+1|t} = F(x_t) \hat{\xi}_{t|t-1} + F(x_t) P_{t|t-1} H(x_t) \{ [H(x_t)]' P_{t|t-1} H(x_t) + R(x_t) \}^{-1} \times \{ y_t - a(x_t) - [H(x_t)]' \hat{\xi}_{t|t-1} \} \]  

\[ P_{t+1|t} = F(x_t) P_{t|t-1} F(x_t)' - \{ F(x_t) P_{t|t-1} H(x_t) \{ [H(x_t)]' P_{t|t-1} H(x_t) + R(x_t) \}^{-1} \times [H(x_t)]' P_{t|t-1} [F(x_t)'] \} + Q(x_t) \]

With time-varying parameter matrices, s-period-ahead forecasts of \( y_{t+s} \) or \( \xi_{t+s} \) for \( s > 1 \) are complex to calculate when \( F, H \) or \( A \) vary stochastically. If \( v_t \) and \( w_t \) are normal, then the one-period-ahead forecasts \( \hat{\xi}_{t+1|t} \) and \( \hat{y}_{t+1|t} \) no longer have the interpretation as linear projections, since equation (6.19) is nonlinear in \( x_t \).

The time-varying coefficient regression model is an important application of a state-space representation with data-dependent parameter matrices.

\[ y_t = x_t' \beta_t + w_t \]  

(6.20)

where \( \beta_t \) is a vector of regression coefficients that is assumed to evolve over time according to

\[ (\beta_{t+1} - \bar{\beta}) = F(\beta_t - \bar{\beta}) + v_{t+1} \]  

(6.21)

Assuming the eigenvalues of \( F \) are all inside the unit circle, \( \bar{\beta} \) has the interpretation as the average or state-space coefficient vector. Equation(6.21) is merely the state equation with \( \xi_t = (\beta_t - \bar{\beta}) \). Equation(6.20) can be re-written as

\[ y_t = x_t' \bar{\beta} + x_t' \xi_t + w_t \]

which is recognised as the form of the observation equation, with \( a(x_t) = x_t' \bar{\beta} \) and [\( H(x_t) \)]' = \( x_t \). Higher-order dynamics for \( \beta_t \) can be included by defining \( \xi_t = [(\beta_t - \bar{\beta})', (\beta_{t-1} - \bar{\beta})', ..., (\beta_{t-p+1} - \bar{\beta})'] \) as in Nicholls and Pagan (1985, pg 437).
We now proceed to implement the Kalman filter to model our foreign exchange data. EViews allow for state space models to be estimated using a powerful recursive algorithm known as the Kalman filter. We apply the Kalman filter to model the daily DEM/USD returns series, the daily ZAR/USD returns series, as well as the intraday (5 minute observations) ZAR/USD returns series.

6.3 State-Space Modelling in EViews using the Kalman filter

6.3.1 Modelling the DEM/USD Returns

We attempt to model the daily DEM/USD exchange rate returns time series using the Kalman filter in EViews. The data set consists of the DEM/USD log return prices \( r_t = p_t - p_{t-1} = \ln P_t - \ln P_{t-1} \) where \( P_t \) is the exchange rate at time period \( t \). The data consists of daily data (Monday to Friday) from 02/01/1990 until 31/12/1998.

After analysis from Chapter 3 and generation of the residuals, we estimate an ARMA(1,1) state space model for the DEM/USD returns. Recall that the ARMA(1,1) model is denoted as

\[ Y_t = \phi_1 Y_{t-1} + \nu_t + \theta_1 \nu_{t-1} \]

The following estimation output is generated by EViews:
<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Std. Error</th>
<th>t-Statistic</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(1)</td>
<td>2.89E-05</td>
<td>0.000114</td>
<td>0.252172</td>
</tr>
<tr>
<td>C(2)</td>
<td>-0.003529</td>
<td>1.695380</td>
<td>-0.002062</td>
</tr>
<tr>
<td>C(3)</td>
<td>0.012617</td>
<td>1.684538</td>
<td>0.007445</td>
</tr>
</tbody>
</table>

Log Likelihood: 12198.82

\[ Y = C(1) + SV1 + C(2) * SV1(-1) \]
\[ SV1 = C(3) * SV1(-1) \]

| R-squared | 0.003343 | Mean dependent var | 2.33E-05 |
| Adjusted R-squared | 0.003048 | S.D. dependent var | 0.006605 |
| S.E. of regression | 0.006595 | Sum squared resid | 0.147177 |
| Durbin-Watson stat | 2.000435 |                  |           |

Figure 6-1: Kalman Filter Estimation Output for DEM/USD returns - ARMA(1,1) model

From Figure (6-1) above we observe the following:

1. The hypothesis that the coefficients C(1), C(2) and C(3) are zero, cannot even be rejected at the 10% level.

2. The R-squared value of 0.003343 is very poor.

3. The standard error of regression is fairly acceptable at 0.006595.

Figure (6-2) below illustrates the actual and one-step ahead fitted values of the signal dependent variable, y, as well as the one-step ahead standardised residuals.
Figure 6-2: Plotted DEM/USD Actual and Predicted Signals and Residuals

The graph of the actual and one-step ahead fitted values of the signal dependent variable, \( y \), and the one-step ahead standardised residuals further confirm that the ARIMA(1,1) model fitted to the daily DEM/USD returns is not a satisfactory model. Residual values are too high and predicted values are insignificant.

We now attempt a different approach to model the daily Deutsche Mark/US Dollar (DEM/USD) exchange rate returns time series using the Kalman filter in EViews. Instead of modelling the DEM/USD exchange rate returns time series as a single variable, we now include explanatory variables such as the Japanese Yen/US Dollar, Gold Dollar Price, SP500 Index (US) and the Dax Index (German). The data set comprises:

- \( Y_t = \) Daily Deutsche Mark/US Dollar (DEM/USD) exchange rate log returns
- \( z_{1t} = \) Daily Japanese Yen/US Dollar (JPY/USD) exchange rate log returns
- \( z_{2t} = \) Daily Dax Index price log returns
- \( z_{3t} = \) Daily S&P500 Index price log returns
- \( z_{4t} = \) Daily Gold price (US Dollars) log returns

The data consists of daily data (Monday to Friday) from 02/01/1990 until 31/12/1998 and each sub data set has 2321 observations. Because the introduction of the Euro occurred in 1999, we choose our data set to exclude observations thereafter. In our state-space model we have
chosen $Y_t$ as the dependent (response) variable, and the explanatory variables $z_{1t}, z_{2t}, z_{3t}, z_{4t}$ are considered to be time-varying parameter variables. As in the case of the neural network computations, the explanatory variables chosen here were based on availability of data. These explanatory variables are a subset of the input factors used by Zimmermann, Grothmann and Neuneier in their research on multi-agent modelling of FX-markets by neural networks.

We first fit a state-space model using ARMA(1,1) estimation. EViews generates the following estimation output:

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Std. Error</th>
<th>t-Statistic</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(1)</td>
<td>0.442919</td>
<td>0.011376</td>
<td>38.93426</td>
</tr>
<tr>
<td>C(2)</td>
<td>0.056604</td>
<td>0.008626</td>
<td>6.550537</td>
</tr>
<tr>
<td>C(3)</td>
<td>0.022074</td>
<td>0.014191</td>
<td>1.555446</td>
</tr>
<tr>
<td>C(4)</td>
<td>-0.018139</td>
<td>0.015552</td>
<td>-1.166373</td>
</tr>
<tr>
<td>C(5)</td>
<td>3.26E-06</td>
<td>0.000126</td>
<td>0.025807</td>
</tr>
<tr>
<td>C(6)</td>
<td>0.241119</td>
<td>0.467139</td>
<td>0.516161</td>
</tr>
<tr>
<td>C(7)</td>
<td>-0.205637</td>
<td>0.470235</td>
<td>-0.437307</td>
</tr>
</tbody>
</table>

Log Likelihood 6666.060

\[ Y = C(1)Z_1 + C(2)Z_2 + C(3)Z_3 + C(4)Z_4 + C(5)SV_1 + C(6)SV_1(-1) + C(7)SV_1(-1) \]

R-squared 0.255767 Mean dependent var -1.12E-05
Adjusted R-squared 0.254159 S.D. dependent var 0.006723
S.E. of regression 0.005806 Sum squared resid 0.078007
Durbin-Watson stat 2.003875

Figure 6-3: Kalman Filter Estimation Output for DEM/USD multi-vector ARMA(1,1) model

From Figure(6-3) above we note the following:

1. The hypothesis that the coefficients $C(1)$ and $C(2)$ are zero, is rejected at the 5% significance level as well as at the 1% level. Therefore coefficients $C(1)$ and $C(2)$ are significant. However, coefficients $C(3),...,C(7)$ cannot even be rejected at the 10% level.

2. The R-squared statistic of 0.255767 is not very good, but certainly better than the previous model.

3. The standard error of regression is reasonably small at 0.005806.
We now assume that the explanatory variables (the z-variables) have time-varying parameters. So, next we fit a time varying random coefficients model with random walk errors, that is

$$\beta_t = \beta_{t-1} + w_t \quad t = 1, ..., T \quad w_t \sim N(0, \sigma_w^2 I)$$

The estimation output from EViews is inserted below:

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Std. Error</th>
<th>t-Statistic</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObVar(1,1)</td>
<td>9.04E-07</td>
<td>52.05766</td>
<td>1.74E-08</td>
</tr>
<tr>
<td>SSVar(1,1)</td>
<td>0.018920</td>
<td>0.000725</td>
<td>26.11204</td>
</tr>
<tr>
<td>SSVar(2,2)</td>
<td>0.023063</td>
<td>0.000788</td>
<td>29.25138</td>
</tr>
<tr>
<td>SSVar(3,3)</td>
<td>0.020184</td>
<td>0.000667</td>
<td>20.89178</td>
</tr>
<tr>
<td>SSVar(4,4)</td>
<td>0.025839</td>
<td>0.000871</td>
<td>29.67648</td>
</tr>
<tr>
<td>SSVar(5,5)</td>
<td>6.35E-17</td>
<td>6.52E+11</td>
<td>7.45E-29</td>
</tr>
<tr>
<td>Final SV1</td>
<td>-0.022538</td>
<td>0.208628</td>
<td>-0.107901</td>
</tr>
<tr>
<td>Final SV2</td>
<td>-0.242161</td>
<td>0.275104</td>
<td>-0.880251</td>
</tr>
<tr>
<td>Final SV3</td>
<td>-0.744429</td>
<td>0.248790</td>
<td>-2.992205</td>
</tr>
<tr>
<td>Final SV4</td>
<td>-1.005327</td>
<td>0.457976</td>
<td>-2.195152</td>
</tr>
<tr>
<td>Final SV5</td>
<td>-5.96E-05</td>
<td>4.31E-05</td>
<td>-1.383118</td>
</tr>
</tbody>
</table>

Log Likelihood 4743.012

Y = SV1*Z1 + SV2*Z2 + SV3*Z3 + SV4*Z4 + SV5
SV1 = SV1(-1)
SV2 = SV2(-1)
SV3 = SV3(-1)
SV4 = SV4(-1)
SV5 = SV5(-1)

R-squared 0.959990 Mean dependent var -1.39E-06
Adjusted R-squared 0.960007 S.D. dependent var 0.006710
S.E. of regression 0.001342 Sum squared resid 0.004170
Durbin-Watson stat 1.969737

Figure 6-4: Kalman Filter Estimation Output for DEM/USD - Time Varying Random Coefficients with RW Errors Model

From the estimation output above, we observe the following:

1. "OBVAR" and "SSVAR" are the elements of the estimated covariance matrix of the
observation and state equations.

2. There is a remarkable increase in the R-squared value, and a value of 0.959990 indicates a very good model fit indeed.

3. In addition the standard error of regression of 0.001342 is very small, almost 5 times smaller than the previous model.

Figure(6-5) below displays the actual and one-step ahead fitted values of the signal dependent variable, y, as well as the one-step ahead standardised residuals.

Figure 6-5: Plotted DEM/USD Actual and Predicted Signals and Residuals

The graph of the actual and one-step ahead fitted values of the signal dependent variable and the one-step ahead standardised residuals indicate that the time varying random coefficients model with random walk errors is indeed a good model fit to the daily DEM/USD returns with explanatory variables daily JPY/USD returns, daily DAX Index returns, daily S&P500 index returns, daily Gold price returns.

We fit another state-space model to the daily DEM/USD returns with explanatory variables, the VAR(1) model without drift, i.e.,
$$\beta_t = \Phi \beta_{t-1} + w_t \quad t = 1, \ldots, T \quad w_t \sim N(0, W_t)$$

$$\Phi = \text{diag}(\phi_1, \ldots, \phi_k)$$

Figure (6-6) below illustrates the estimation output for the VAR(1) model without drift.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Std. Error</th>
<th>t-Statistic</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(1)</td>
<td>-1.017034</td>
<td>-160.9350</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(2)</td>
<td>-0.361645</td>
<td>-8.456448</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(3)</td>
<td>-0.306256</td>
<td>-6.374812</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(4)</td>
<td>0.431473</td>
<td>11.30113</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(5)</td>
<td>0.208719</td>
<td>5.976098</td>
<td>0.0000</td>
</tr>
<tr>
<td>ObVar(1,1)</td>
<td>5.91E-06</td>
<td>4.72E-07</td>
<td>1.0000</td>
</tr>
<tr>
<td>SSVar(1,1)</td>
<td>0.066875</td>
<td>22.61153</td>
<td>0.0000</td>
</tr>
<tr>
<td>SSVar(2,2)</td>
<td>0.019011</td>
<td>6.762737</td>
<td>0.0000</td>
</tr>
<tr>
<td>SSVar(3,3)</td>
<td>0.040149</td>
<td>11.29347</td>
<td>0.0000</td>
</tr>
<tr>
<td>SSVar(4,4)</td>
<td>0.055247</td>
<td>12.24868</td>
<td>0.0000</td>
</tr>
<tr>
<td>SSVar(5,5)</td>
<td>5.54E-15</td>
<td>3.62E-25</td>
<td>1.0000</td>
</tr>
<tr>
<td>Final SV1</td>
<td>-0.358227</td>
<td>-1.079483</td>
<td>0.2605</td>
</tr>
<tr>
<td>Final SV2</td>
<td>-0.008308</td>
<td>-0.056214</td>
<td>0.9652</td>
</tr>
<tr>
<td>Final SV3</td>
<td>-0.006241</td>
<td>-0.029659</td>
<td>0.9763</td>
</tr>
<tr>
<td>Final SV4</td>
<td>0.037150</td>
<td>0.143362</td>
<td>0.8660</td>
</tr>
<tr>
<td>Final SV5</td>
<td>2.74E-13</td>
<td>3.60E-06</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Log Likelihood: 7194.695

Y = SV1*Z1 + SV2*Z2 + SV3*Z3 + SV4*Z4 + SV5
SV1 = C(1)*SV1(-1)
SV2 = C(2)*SV2(-1)
SV3 = C(3)*SV3(-1)
SV4 = C(4)*SV4(-1)
SV5 = C(5)*SV5(-1)

R-squared: 0.804483
Adjusted R-squared: 0.804467
S.E. of regression: 0.002966
Durbin-Watson stat: 2.030964

Figure 6-6: Kalman Filter Estimation Output for DEM/USD - VAR(1) Model without Drift

The estimation output above illustrates the following:
1. The hypothesis that the coefficients C(1),...,C(5) are zero, is rejected at the 5% significance level as well as at the 1% level. Therefore coefficients C(1),...,C(5) are all significant.

2. The R-squared value of 0.804483 is less than the previous model.

3. The standard error of regression of 0.002966 is larger than the previous model.

This model should then be discarded in favour of the previous model.

6.3.2 Modelling the ZAR/USD Returns

Modelling the ZAR/USD Daily Returns

Next we attempt to model the daily ZAR/USD exchange rate returns time series. The data set consists of the ZAR/USD log return prices. The data consists of 3387 daily observations (Monday to Friday) from 02/01/1990 until 31/12/2002.

The estimation output generated by an ARMA(1,1) state-space model follows:

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Std. Error</th>
<th>t-Statistic</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(1)</td>
<td>0.000363</td>
<td>3.139977</td>
<td>0.0017</td>
</tr>
<tr>
<td>C(2)</td>
<td>0.574581</td>
<td>-3.689960</td>
<td>0.0002</td>
</tr>
<tr>
<td>C(3)</td>
<td>0.542801</td>
<td>3.362357</td>
<td>0.0008</td>
</tr>
</tbody>
</table>

Log Likelihood 12019.80

\[
Y = C(1) + SV1 + C(2)SV1(-1) \\
SV1 = C(3)SV1(-1)
\]

<table>
<thead>
<tr>
<th>R-squared</th>
<th>0.003758</th>
<th>Mean dependent var</th>
<th>0.000363</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjusted R-squared</td>
<td>0.003462</td>
<td>S.D. dependent var</td>
<td>0.006776</td>
</tr>
<tr>
<td>S.E. of regression</td>
<td>0.006765</td>
<td>Sum squared resid</td>
<td>0.153661</td>
</tr>
<tr>
<td>Durbin-Watson stat</td>
<td>1.964066</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6-7: Kalman Filter Estimation Output for ZAR/USD returns - ARMA(1,1) model

From the figure above we observe the following:

1. The hypothesis that the coefficients C(1),...,C(3) are zero, is rejected at the 5% significance level as well as at the 1% level. Therefore coefficients C(1),...,C(3) are all statistically
2. The R-squared value of 0.003758 is too small.

3. The standard error of regression is fairly satisfactory at 0.006765.

Figure (6-8) below displays the actual and one-step ahead fitted values of the signal dependent variable, \( y \), as well as the one-step ahead standardised residuals.

![Graph of the actual and one-step ahead fitted values of the signal dependent variable, y, with actual and predicted values.](image)

**Figure 6-8: Plotted ZAR/USD Actual and Predicted Signals and Residuals**

The graph of the actual and one-step ahead fitted values of the signal dependent variable, \( y \), and the one-step ahead standardised residuals further confirm that the ARMA(1,1) model fitted to the daily ZAR/USD returns is not a suitable model. This poor result is similar to the attempt to model the daily DEM/USD returns as a single variable.

Similarly, as in the case of the DEM/USD returns, we attempt to model the daily SA Rand/US Dollar (ZAR/USD) returns by including explanatory variables such as the British Pound/US Dollar, Swiss Franc/US Dollar, Japanese Yen/US Dollar, and Gold Dollar Price. These explanatory variables were used in our multi-agent modelling of the SA FX-market by neural networks. The data set comprises:

\[ y_t = \text{Daily SA Rand/US Dollar (ZAR/USD) exchange rate log returns} \]
\[ z_{1t} = \text{Daily British Pound/US Dollar (GBP/USD) exchange rate log returns} \]
\[ z_{2t} = \text{Daily Swiss Franc/US Dollar (CHF/USD) exchange rate log returns} \]
\[ z_{3t} = \text{Daily Japanese Yen/US Dollar (JPY/USD) exchange rate log returns} \]
\[ z_{4t} = \text{Daily Gold price (US Dollars) log returns} \]

The data consists of daily data (Monday to Friday) from 02/01/1990 until 31/12/2002 and each sub data set has 3361 observations. In our state-space model we have chosen \( Y \) as the dependent (response) variable, and the explanatory variables \( z_{1t}, z_{2t}, z_{3t}, z_{4t} \) are considered to be time-varying parameter variables.

We first model the SA FX-market by fitting a state-space model using ARMA(1,1) estimation. The estimation output obtained from EViews is as follows:

<table>
<thead>
<tr>
<th></th>
<th>Coefficient</th>
<th>Std. Error</th>
<th>t-Statistic</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(1)</td>
<td>0.000366</td>
<td>0.000111</td>
<td>3.295531</td>
<td>0.0010</td>
</tr>
<tr>
<td>C(2)</td>
<td>-0.111764</td>
<td>0.028799</td>
<td>-3.880791</td>
<td>0.0001</td>
</tr>
<tr>
<td>C(3)</td>
<td>0.099842</td>
<td>0.021460</td>
<td>4.686452</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(4)</td>
<td>0.050569</td>
<td>0.012477</td>
<td>4.054521</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(5)</td>
<td>-0.062096</td>
<td>0.013370</td>
<td>-4.644594</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(6)</td>
<td>-0.548974</td>
<td>0.110229</td>
<td>-4.890293</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(7)</td>
<td>0.501128</td>
<td>0.115991</td>
<td>4.320407</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Log Likelihood 12107.66

\[ Y = C(1) + C(2)z_1 + C(3)z_2 + C(4)z_3 + C(5)z_4 + SV1 + C(6)SV1(-1) \]
\[ SV1 = C(7)SV1(-1) \]

R-squared 0.054525 Mean dependent var 0.000358
Adjusted R-squared 0.053116 S.D. dependent var 0.006776
S.E. of regression 0.008694 Sum squared resid 0.145830
Durbin-Watson stat 1.980545

Figure 6-9: Kalman Filter Estimation Output for ZAR/USD multi-vector ARMA(1,1) model

From the estimation output above the following is noted:

1. The hypothesis that the coefficients \( C(1), \ldots, C(7) \) are zero, is rejected at the 5% significance level as well as at the 1% level. Therefore coefficients \( C(1), \ldots, C(7) \) are all significant.

2. The Durbin-Watson statistic of 1.980545 is fairly close to 2 so we can assume not much
serial correlation exists.

3. The standard error of regression of 0.006594 is relatively small.

4. The R-squared value of 0.050333 seems too small.

Next we fit a time varying random coefficients model with random walk errors. The estimation output from EViews is inserted below:

<table>
<thead>
<tr>
<th></th>
<th>Coefficient</th>
<th>Std. Error</th>
<th>t-Statistic</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObVar(1,1)</td>
<td>1.73E-05</td>
<td>1.900065</td>
<td>9.13E-06</td>
<td>1.0000</td>
</tr>
<tr>
<td>SSVar(1,1)</td>
<td>1.03E-21</td>
<td>3.65E+16</td>
<td>2.83E+38</td>
<td>1.0000</td>
</tr>
<tr>
<td>SSVar(2,2)</td>
<td>0.044146</td>
<td>0.003600</td>
<td>12.26246</td>
<td>0.0000</td>
</tr>
<tr>
<td>SSVar(3,3)</td>
<td>0.044065</td>
<td>0.002472</td>
<td>17.82359</td>
<td>0.0000</td>
</tr>
<tr>
<td>SSVar(4,4)</td>
<td>0.042076</td>
<td>0.002226</td>
<td>18.89980</td>
<td>0.0000</td>
</tr>
<tr>
<td>SSVar(5,5)</td>
<td>0.040382</td>
<td>0.003436</td>
<td>11.75143</td>
<td>0.0000</td>
</tr>
<tr>
<td>Final SV1</td>
<td>0.000330</td>
<td>9.21E-05</td>
<td>3.583719</td>
<td>0.0003</td>
</tr>
<tr>
<td>Final SV2</td>
<td>0.131855</td>
<td>0.637293</td>
<td>2.068899</td>
<td>0.0386</td>
</tr>
<tr>
<td>Final SV3</td>
<td>-0.199154</td>
<td>0.515313</td>
<td>-0.366473</td>
<td>0.6992</td>
</tr>
<tr>
<td>Final SV4</td>
<td>1.756060</td>
<td>0.537428</td>
<td>3.271247</td>
<td>0.0011</td>
</tr>
<tr>
<td>Final SV5</td>
<td>-0.584786</td>
<td>0.362877</td>
<td>-1.611528</td>
<td>0.1072</td>
</tr>
</tbody>
</table>

Log Likelihood    | 11359.27

\[ Y = SV1 + SV2*Z1 + SV3*Z2 + SV4*Z3 + SV5*Z4 \]
\[ SV1 = SV1(-1) \]
\[ SV2 = SV2(-1) \]
\[ SV3 = SV3(-1) \]
\[ SV4 = SV4(-1) \]
\[ SV5 = SV5(-1) \]

R-squared        | 0.630913   |
Adjusted R-squared| 0.631023   |
S.E. of regression| 0.004113   |
Durbin-Watson stat| 1.994139   |

Figure 6-10: Kalman Filter Estimation Output for ZAR/USD - Time Varying Random Coefficients with RW Errors Model

We observe the following from the estimation output in figure(6-10) above:
1. The R-squared statistic of 0.630913 indicates that the regression model is fairly good in predicting the values of the dependent variable within the sample. The R-squared is much higher than in the case of fitting an ARMA(1,1) model.

2. The Durbin-Watson statistic of 1.994139 is fairly close to 2 so we can assume not much serial correlation exists.

3. The standard error of regression of 0.004113 is therefore smaller than the standard error of regression in the previous state-space estimation.

4. A remarkable improvement in the model fit is noted.

We now try to improve upon this state-space model by introducing a VAR(1) model without drift. The new estimation output is inserted below.
<table>
<thead>
<tr>
<th></th>
<th>Coefficient</th>
<th>Std. Error</th>
<th>t-Statistic</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(1)</td>
<td>-0.156138</td>
<td>0.013095</td>
<td>-11.92371</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(2)</td>
<td>0.917575</td>
<td>0.005319</td>
<td>172.5001</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(3)</td>
<td>-0.909389</td>
<td>0.004114</td>
<td>-221.0372</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(4)</td>
<td>-0.101257</td>
<td>0.036956</td>
<td>-2.739939</td>
<td>0.0062</td>
</tr>
<tr>
<td>C(5)</td>
<td>0.966597</td>
<td>0.003317</td>
<td>297.4489</td>
<td>0.0000</td>
</tr>
<tr>
<td>ObVar(1,1)</td>
<td>5.94E-06</td>
<td>234.9484</td>
<td>2.53E-08</td>
<td>1.0000</td>
</tr>
<tr>
<td>SSVar(1,1)</td>
<td>4.72E-06</td>
<td>296.6205</td>
<td>1.59E-08</td>
<td>1.0000</td>
</tr>
<tr>
<td>SSVar(2,2)</td>
<td>0.064064</td>
<td>0.003769</td>
<td>16.99578</td>
<td>0.0000</td>
</tr>
<tr>
<td>SSVar(3,3)</td>
<td>0.023001</td>
<td>0.001740</td>
<td>13.21517</td>
<td>0.0000</td>
</tr>
<tr>
<td>SSVar(4,4)</td>
<td>0.033536</td>
<td>0.001804</td>
<td>16.59215</td>
<td>0.0000</td>
</tr>
<tr>
<td>SSVar(5,5)</td>
<td>0.027140</td>
<td>0.001559</td>
<td>17.40778</td>
<td>0.0000</td>
</tr>
<tr>
<td>Final SV1</td>
<td>0.000316</td>
<td>0.002194</td>
<td>0.143996</td>
<td>0.8856</td>
</tr>
<tr>
<td>Final SV2</td>
<td>-0.292262</td>
<td>0.494505</td>
<td>-0.591020</td>
<td>0.5545</td>
</tr>
<tr>
<td>Final SV3</td>
<td>-0.406579</td>
<td>0.296249</td>
<td>-1.379175</td>
<td>0.1679</td>
</tr>
<tr>
<td>Final SV4</td>
<td>0.007708</td>
<td>0.184062</td>
<td>0.041878</td>
<td>0.9666</td>
</tr>
<tr>
<td>Final SV5</td>
<td>0.119098</td>
<td>0.260258</td>
<td>0.457617</td>
<td>0.6473</td>
</tr>
</tbody>
</table>

Log Likelihood = 11025.58

\[ Y = SV1 +SV2*Z1 +SV3*Z2 +SV4*Z3 +SV5*Z4 \]

SV1 = C(1)*SV1(-1)
SV2 = C(2)*SV2(-1)
SV3 = C(3)*SV3(-1)
SV4 = C(4)*SV4(-1)
SV5 = C(5)*SV5(-1)

- R-squared = 0.890005
- Mean dependent var = 0.000362
- Adjusted R-squared = 0.890037
- S.D. dependent var = 0.006772
- S.E. of regression = 0.002246
- Sum squared resid = 0.016922
- Durbin-Watson stat = 1.871053

Figure 6-11: Kalman Filter Estimation Output for ZAR/USD - VAR(1) Model without Drift

From the estimation output above we note the following:

1. The hypothesis that the coefficients C(1),...,C(5) are zero, is rejected at the 5% significance level and at the 1% level for coefficients C(1),C(2),C(3),C(5). Therefore coefficients C(1),...,C(5) are all significant.

2. The R-squared statistic of 0.890005 is higher than the previous model.

3. The standard error of regression of 0.002246 is also smaller than the previous model.
Figure (6-12) below displays the actual and one-step ahead fitted values of the signal dependent variable, \( y \), as well as the one-step ahead standardised residuals.

![One-step-ahead Y](image)

Figure 6-12: Plotted ZAR/USD Actual and Predicted Signals and Residuals.

The periods of high volatility seem to affect model predictive abilities, for approximately the first third of the time series the residuals are relatively smaller than residuals in the high volatility periods in the time series.

**Modelling the ZAR/USD Intraday (5-minute) Returns**

We apply the Kalman filter to the very high frequency (5 minute intervals) ZAR/USD exchange rate returns time series. The data set consists of the ZAR/USD log return prices. The data comprises 20592 observations (Monday to Friday) from 01/07/2002 until 31/01/2003.

The estimation output for the ARMA(1,1) is shown below.
Figure 6-13: Kalman Filter Estimation Output for ZAR/USD 5 min returns - ARMA(1,1) model

From Figure(6-13) above, the following is observed:

1. The hypothesis that the coefficient C(2) is zero, is rejected at the 5% significance level as well as at the 1% level. Therefore coefficient C(2) is statistically significant. However for C(1) and C(3) we cannot reject the null hypothesis.

2. The R-squared value of 0.083367 is not satisfactory.

3. The standard error of regression is relatively small at 0.001186.

Figure(6-14) below displays the actual and one-step ahead fitted values of the signal dependent variable, y, as well as the one-step ahead standardised residuals.
Predictive abilities of the model appear relatively good with the residuals remaining consistently small. The predicted values appear to be slightly offset from the actual values.

### 6.4 Concluding Remarks

Based on the analysis and results above, the following conclusions may be drawn:

1. The daily DEM/USD and daily ZAR/USD returns are better modelled within the state-space framework using multiple explanatory variables as opposed to being modelled using a single variable.

2. For the daily DEM/USD returns, the state-space model that produces the best estimates is the time varying random coefficients model with random walk errors.

3. For the daily ZAR/USD returns, the state-space model that produces the best estimates is the VAR(1) model without drift.

4. A possible recommendation would be to model the ZAR/USD returns using a Kalman filter that is adapted to a non-linear state-space model, or a state-space model that has...
non-normal noise. It would be useful to compare those results to the EViews linear state-space model results.

A brief description of these Kalman filter models follows.

The Extended Kalman Filter

As discussed previously, the Kalman filter addresses the general problem of trying to estimate the state of a discrete-time controlled process that is governed by a linear stochastic difference equation. What then happens if the process to be estimated and/or the measurement relationship to the process is non-linear? A Kalman filter that linearises about the current mean and covariance is referred to as an extended Kalman filter (EKF).

The Gaussian estimation methods can produce very poor results in the presence of gross outliers. The Kalman filter can be adapted to filter processes where the disturbance terms are non-Gaussian. If the distribution is known, then the optimal minimum mean variance estimator can be calculated. The Kalman filter can be extended to filter non-linear state-space models. An approximate filter is used, since these models are not generally conditionally Gaussian. The state space model’s observation function and the state update function are no longer linear functions of the state vector.

Conditionally Gaussian models have Gaussian disturbances but allow the system matrices to be stochastic, in that they depend on information (observations) available at time \( t - 1 \). The Kalman filter can still be used to model these conditionally Gaussian models. If observations in the measurement equation are no longer a linear function of the state vector, and in the transition equation, the state vector itself is no longer a linear function of the state vector in the previous period, then the model is functionally non-linear. Since these models are, in general, not conditionally Gaussian it is necessary to resort to approximate filters, the most basic one being the extended Kalman filter. Some approximations are necessary for the estimation procedure, since explicit expressions for the filtering algorithms cannot be derived.

There are two main approaches to obtain non-linear filtering algorithms. The first approach is to approximate the non-linear measurement and transition equations. The linearised non-linear functions are then applied to a modification of the linear Kalman filter algorithm. The other type of algorithms can be summarized under the approach of approximating the
underlying density functions of the state vector. A recursive algorithm on the densities is then derived using Bayes' formula. This approach is advantageous in that it results in asymptotically unbiased filtering estimates. However, the estimators based on the density approach require a great amount of computational burden compared with those based on the Taylor series approximations (Tanizaki, 1996).

**Unscented Kalman Filter**

The unscented Kalman filter is accurate to 3rd order, for a Gaussian process and measurement noise, assuming that the prior state is also Gaussian. For non-Gaussian distributions, it is accurate to at least 2nd order. Hence, the unscented Kalman filter offers a better performance than the extended Kalman filter. The unscented Kalman filter is simpler in implementation, because the unscented Kalman filter does not require the computation of Jacobian, unlike the extended Kalman filter. It can also be applied to non-differentiable nonlinear dynamical systems.

**Particle Filters**

Extended Kalman filters and unscented Kalman filters rely on the use of approximations to ensure mathematical tractability. Particle filters provide a tool for recursive computation of a stochastic point-mess approximation to the posterior distribution of the hidden states of a nonlinear dynamical system, given a set of observations related to the states. Particle filtering aims to recursively estimate the posterior distribution, the filtering distribution, and specific expectations such as the conditional mean and conditional covariance.

In Monte Carlo simulation, a set of weighted, independent and identically distributed particles (i.e. random samples) is drawn from the posterior distribution, thereby mapping integrals to discrete sums. Often, it is not possible to draw samples directly from the posterior distribution, and hence implement an arbitrary proposal distribution to formulate the importance sampling. We could use approximations based on the extended Kalman filter or unscented Kalman filter to generate Gaussian proposal distributions. Importance sampling does not lend itself to recursive estimation, which is the computational goal of sequential state estimation. To solve this problem, we use sequential importance sampling as it yields a set of parameters
known as normalized importance weights.

Thus far, we have not modelled the very high frequency (5 minute observations) ZAR/USD data using explanatory variables, as the explanatory variables used in the Kalman filter estimation and multi-agent neural network approach were available at a daily frequency only. In the next chapter, we employ fuzzy Markov chains to model the 5 minute ZAR/USD observations and attempt a short-term prediction.
Chapter 7

FUZZY MARKOV CHAIN
MODELLING OF HIGH
FREQUENCY FOREX DATA

The aim of this chapter is to utilise fuzzy Markov chains to model very high frequency forex data as stated in the research objectives.

7.1 Fuzzy Uncertainty

It is a well-known fact that change is a fundamental fact of the universe and society, and therefore not every component in a stated scientific law will last forever. Any scientific law and methodology are inevitably evolving because science and technology are evolving and advancing from a low or primal level to a high or natural level. However, no matter what the form of change, the basic direction is that the new or updated scientific law enjoys a more objective, deeper insight into the nature or reality of the universe and society. As to econometric modelling there should be continuous updating, reshaping and even reforming. Nowadays, it often appears that econometric model developments do not sufficiently reflect the underlying market information. Consequently these economic and econometric models do not approach the financial market realities well enough, and financial practitioners can neither apply those models nor obtain fruitful results in their applications. Therefore it is necessary to address the
root causes of the imbalance between econometric modelling developments and their practical applications.

One of the reasons behind financial market models moving away from financial practice may root deeply in the modern scientific methodology itself. Just as Alvin Toffler pointed out in the book “Order out of Chaos” (I. Prigogine and I. Stengers, 1984), “One of the most highly developed skills in contemporary Western civilization is dissection: the split-up of problems into their smallest possible components. We are good at it. So good, we often forget to put the pieces back together again.” In other words, it is often the practice in scientific modelling to isolate the focus point away from its environment, thereby ignoring the complex interactions of the problem studied and the rest of the real world. It is not an easy task to address the problems resulting from dissection methodology. However, two aspects are worth noticing. Firstly, within the modelling process it is critical to identify the fundamental characteristic of the focused problem, since such a fundamental characteristic is common to the problem itself as well as its environment. Secondly, it is an essential task to account for the environmental impacts, i.e., the interactions between the problem studied and its environment.

Uncertainty and certainty are aspects of the fundamental characteristics of the world surrounding us. In human history, certainty is the first form of perception of the living world and often thought of as a basic feature of the existing world and an elemental human thinking pattern. Since the establishment of Newtonian physics, certainty thinking dominated science and society for centuries. Certainty is a phenomenon in nature resulting when all the necessary and sufficient conditions for the occurrence of an event are satisfied. Only in the last century has uncertainty, particularly random uncertainty resulting from partial conditions of an event’s occurrence being satisfied, been understood and received by the scientific and economic community. Prigogine (1997) strongly argued that probability is no longer merely a mathematical methodology in modelling natural processes, but an intrinsic characteristic of natural processes. However, other forms of uncertainty, especially fuzzy uncertainty (the reflection of the unclear boundary of events or evolution processes), was accepted by the scientific community since 1960’s.

Philosophically speaking, both randomness and fuzziness are objectively existing, inherent, and inseparable characteristics of the world’s reality. Firstly, every event occurring in a realistic
world, no matter how complicated it may be, always enjoys its own self-existence and self-specification and therefore it is objective and certain. Secondly, the self-specification associated with each existing object does not often enjoy a clear and defined boundary and the pattern of change can be either evolving or having a sudden jump. However, in general it is not a sudden jump from one level to another but more possibly evolving from one stratum to another. In other words, between two different strata, there exist a few middle layers. At these in-between layers, the event possesses features of various degrees from both strata and therefore vague boundaries and specifications are an objective existing form of our world reality. Thirdly, the occurrence of any event ought to satisfy its generating conditions. If all the related generating conditions are satisfied, then it would be a certainty event; if partial generating conditions are satisfied, then the event will be a chance event. Thus the cause-effect or generalized cause-effect relationship is also an objectively existing form of changes. Fourthly, the true states of the existing real world are never completely grasped by human beings. The information collected reflects what is available about the real world. However, information collection itself is inevitably realised with the brand of times. The instruments and the depth of knowledge about our objective existing world are all limited by the level of science and technology achieved at the time. Therefore, the information collected at any time will have observational errors (random in general) and observational biases (fuzzy in nature). Thus, fuzzy and random uncertainties are intrinsically associated with information data. Therefore in order to correctly utilise data information in a decision-making process, we have to understand the fundamental and intrinsic features of information, that is, uncertainty.

Logically speaking, randomness and fuzziness are two different types of uncertainties. Randomness, which is logically the break down of the law of causality because of the lack of some conditions under which the event occurrence is inevitable, is the traditionally well-received formality of uncertainty in terms of the usage of probability calculus by science and engineering. However, just as L. A. Zadeh (1988) pointed out, "it has become increasingly clear that there are some important facets of uncertainty which do not lend themselves to analysis by classical probability-based methods." Fuzziness, which is logically the break down of the law of excluding the middle, is neither well known nor largely ignored in the community of financial time series modelling.
Covariate modelling received attention since Cox's (1972) milestone paper. Covariate modelling examines at the interactions between the focused problem and its environment. However, the dissection shadow often directs the modelling interactions efforts away from its correct path by isolating the so-called significant covariate variables and ignoring the remaining covariate information which is part of the environment. Therefore in this chapter we intend to address the fatal weakness of dissection methodology by considering the randomness and fuzziness uncertainties, which are a fundamental commonality to any split-up part, and the covariates, which provide the interactions to the environment. However, the way that we address this is synthetic rather than isolating by taking the advantages of the fuzzy probabilistic methodology.

We now review the basic concepts on probability of fuzzy events initiated by Zadeh (1965).

### 7.2 Fuzzy Probability Calculus

Traditionally, the random characteristic in economic and financial information is facilitated by classical probability calculus. Therefore logically the co-existence of fuzziness and randomness in economic and financial information require a fuzzy probability calculus that combines fuzzy and random elements from fuzzy calculus and classical probability calculus. Such a combination would help to lay down a cornerstone of a new calculus to reflect the nature of the dynamics of financial markets.

#### 7.2.1 Probability of Fuzzy Events

The key concept to extend the classical probability calculus towards the fuzzy probability calculus is the indicator function of a random event \( A \in \mathcal{U} \), a \( \sigma \)-algebra of \( \mathcal{U} \)

\[
\vartheta_A (u) = \begin{cases} 
1 & \text{if } \omega \in A \\
0 & \text{otherwise } (\omega \notin A)
\end{cases}
\]

The diagram below illustrates the graphical representation of a random event and the indicator function which characterises the relationship between an element \( \omega \in A \) and set \( A \in \mathcal{U} \).
One fundamental fact is that

$$\Pr(\mathcal{A}) = E\{ \vartheta_\mathcal{A}(X) \} = \int_U \vartheta_\mathcal{A}(u) dP(u)$$

where the right hand side is an abstract Lebesgue integral.

Classical probability calculus requires that random event \( \mathcal{A} \) is a common subset, i.e., for all \( u \in \mathcal{A} \), the belonging relation is definite: it either belongs to \( \mathcal{A} \) or it does not, there is no middle ground. Therefore classical probability calculus cannot describe random events without a clear boundary, i.e., fuzzy events. L. A. Zadeh (1965) defined fuzzy sets in terms of the extension indicator function of a normal subset.

The membership function of a fuzzy set \( \tilde{\mathcal{A}} \) of \( U \) is a mapping from \( U \) onto \([0,1]\)

$$\mu_{\tilde{\mathcal{A}}} : U \rightarrow [0,1].$$

Mapping \( \mu_{\tilde{\mathcal{A}}} \) is called the membership function of \( \tilde{\mathcal{A}} \). The Borel measurable function \( \mu_{\tilde{\mathcal{A}}}(u) \) represents the degree of element \( u \) belonging to set \( \tilde{\mathcal{A}} \). The following graphs should give an intuitive idea of how a fuzzy set and membership function looks.
Thus the probability of fuzzy event $\tilde{A}$ on $\mathcal{U}$, $\tilde{A} \in \tilde{\mathcal{U}}$, is defined as

$$\Pr [\tilde{A}] = E_P [\mu_{\tilde{A}}(X)] = \int_{\tilde{\mathcal{U}}} \mu_{\tilde{A}}(u) dP(u).$$

**Definition 9** Given a probability space $(\mathcal{U}, \mathcal{U}, P)$, let $\tilde{\mathcal{U}}$ be the collection of all the fuzzy events on $\mathcal{U}$, then $(\mathcal{U}, \tilde{\mathcal{U}}, P)$ is called the induced fuzzy probability space from $(\mathcal{U}, \mathcal{U}, P)$. For any fuzzy event $\tilde{A} \in \tilde{\mathcal{U}}$, its probability is

$$\Pr [\tilde{A}] = \int_{\tilde{\mathcal{U}}} \mu_{\tilde{A}}(u) dP(u).$$

It is obvious that the induced Boolean field $\tilde{\mathcal{U}} \supseteq \mathcal{U}$ and further satisfies

(a) $\mathcal{U} \subseteq \tilde{\mathcal{U}}$, $\emptyset \in \tilde{\mathcal{U}}$;

(b) $\tilde{A} \in \tilde{\mathcal{U}} \Rightarrow \tilde{A}^c \in \tilde{\mathcal{U}}$;

(c) $\forall n, \tilde{A}_n \in \tilde{\mathcal{U}} \Rightarrow \bigcup_{n=1}^{\infty} \tilde{A}_n \in \tilde{\mathcal{U}}$, $\bigcap_{n=1}^{\infty} \tilde{A}_n \in \tilde{\mathcal{U}}$.

Therefore, the fuzzy probability calculus can be established naturally as the extension to the classical probability calculus, except

$$\mu_{\tilde{A} \cap \tilde{B}} \triangleq \mu_{\tilde{A}} \cdot \mu_{\tilde{B}}$$
rather than the conventional rule
\[ P(A \cap B) = P(A) \land P(B) \]
to maintain the classical formality of independence, conditional probability, law of total probability as well as Bayes formula.

The intuitive interpretation of the probability of a fuzzy event \( \tilde{A} \) is very similar to that of the probability of an event \( A \) in classical probability theory. The probability of a fuzzy event \( \tilde{A} \) is the average number of \( u \) falling in set \( \tilde{A} \), i.e., the proportion (frequency) of \( u \) falling in \( \tilde{A} \) because a membership of set \( \tilde{A} \) equals to one just means \( u \) falling in fuzzy set \( \tilde{A} \). For the intermediate states, the membership \( 0 < \mu_{\tilde{A}} < 1 \), means \( u \) falls in fuzzy set \( \tilde{A} \) with a grade (degree) of value \( \mu_{\tilde{A}}(u) \). Therefore the proportion of \( u \) falling in fuzzy set \( \tilde{A} \) is an average number of falling in weighing over the membership \( \mu_{\tilde{A}}(\cdot) \).

7.2.2 Relative membership Degree

There is not necessarily intermediary evolution between two different objectively existing events. The occurrence of an intermediary evolution requires a particular condition. The basic condition for an intermediary evolution between two events is that there is a commonality in nature as the common dimensionality. Only two different events having the co-dimensionality could facilitate an intermediary evolution. Therefore, the fuzziness is a characteristic of the belonging to this and belonging to that at the same time revealed when events existing objectively, or concepts are in an intermediary evolution stage under the difference of the co-dimensionality. The concept of membership functions in fuzzy set theory is used to describe the intermediary evolution and thus it is the foundation upon which fuzzy theory is built. However, the definition given by Zadeh (1965) is simply an extension of the set indicator function in classical set theory thus making the membership concept absolute. It does not completely reflect the most fundamental characteristic - intermediary evolution or belonging to this and belonging to that at the same time. Such an absolute definition was inevitably affecting the wide applications thereby creating unnecessary controversies. Therefore it is meaningful to update the definition of the (absolute) membership function to better reflect the fundamental characteristic of fuzzy sets.
Definition 10 (S Y Chen, 1998) Let $\tilde{A}$ be a fuzzy subset (or a fuzzy concept) on the universe of discourse $U$, and number $0$ and $1$ be assigned to the two poles of an intermediary evolution of co-dimensionality with $\tilde{A}$ (on $U$). Let $C[0, 1]$ be a continuum on closed interval $[0, 1]$. For $\forall u \in U$, there exists a number $\mu_{\tilde{A}}^{abs}(u) \in C[0, 1]$. $\mu_{\tilde{A}}^{abs}(u)$ is called an absolute membership of $u$ with respect to $\tilde{A}$. Mapping

$$
\mu_{\tilde{A}}^{abs}: U \rightarrow [0, 1] \\
u \mapsto \mu_{\tilde{A}}^{abs} \in [0, 1] 
$$

is called an absolute membership function.

Definition 11 (S Y Chen, 1998) Assume that a reference system is set up based on the continuum $C[0, 1]$ and any two arbitrary points of $C[0, 1]$ are regarded as two poles and therefore assigned number $0$ and $1$ and that a reference continuum $C^{ref}[0, 1]$ is constructed on the $[0, 1]^{ref}$ reference system. For $\forall u \in U$, there exists a number $\mu_{\tilde{A}}^{rel}(u) \in C^{ref}[0, 1]$. $\mu_{\tilde{A}}^{rel}(u)$ is called a relative membership of $u$ with respect to $\tilde{A}$. Mapping

$$
\mu_{\tilde{A}}^{rel}: U \rightarrow [0, 1]^{ref} \\
u \mapsto \mu_{\tilde{A}}^{rel} \in [0, 1]^{ref} 
$$

is called a relative membership function.

Chen's updated definitions, particularly the proposed relative membership degree concept, offer a deeper insight into the notion of membership and thus greater convenience in engineering and financial applications. In the illustrative example section, computation aspects of the relative membership degree are explained in detail.

7.2.3 Sample Membership for ZAR/USD 5 minute data

The high frequency ZAR/USD data sample utilized in this chapter is taken from 01/07/02 until 31/01/2003. Simple analysis of the data yields,
<table>
<thead>
<tr>
<th>Number of log-returns</th>
<th>20583</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum</td>
<td>$u_{\text{max}} = 0.014779$</td>
</tr>
<tr>
<td>Upper 75 percentile</td>
<td>$u_{0.75} = 0.060359$</td>
</tr>
<tr>
<td>Median</td>
<td>$u_{0.5} = 0.000000$</td>
</tr>
<tr>
<td>Lower 75 percentile</td>
<td>$u_{0.25} = -0.060590$</td>
</tr>
<tr>
<td>Minimum</td>
<td>$u_{\text{min}} = -0.014169$</td>
</tr>
<tr>
<td>Average</td>
<td>$-8.74861E-06$</td>
</tr>
</tbody>
</table>

Then, fit a probability distribution by deleting any value that is less than $u_{0.25}$ and greater than $u_{0.75}$. We observe that the log-returns falling within the interquartile range, i.e., $u_{0.25} \leq u \leq u_{0.75}$ should fit a fairly "normal" distribution.

The fuzzy partition should be $\mathbf{P} = \{E_1, E_2, E_3\}$ with membership functions:

\[
\mu_{E_1}(u) = \begin{cases} 
1 & \text{if } u_{\text{min}} \leq u < u_{0.25} \\
\frac{u_{0.25} - u}{u_{0.25} - u_{0.5}} & \text{if } u_{0.25} \leq u < u_{0.5} \\
0 & \text{if } u \leq u_{\text{min}} \text{ or } u > u_{0.5} 
\end{cases}
\]

\[
\mu_{E_2}(u) = \begin{cases} 
\frac{u - u_{0.25}}{u_{0.5} - u_{0.25}} & \text{if } u_{0.25} \leq u < u_{0.5} \\
\frac{u_{0.75} - u}{u_{0.75} - u_{0.5}} & \text{if } u_{0.5} \leq u < u_{0.75} \\
0 & \text{otherwise} 
\end{cases}
\]

\[
\mu_{E_3}(u) = \begin{cases} 
1 & \text{if } u_{0.75} \leq u \leq u_{\text{max}} \\
\frac{u - u_{0.75}}{u_{\text{max}} - u_{0.75}} & \text{if } u_{0.5} < u < u_{0.75} \\
0 & \text{if } u < u_{0.5} \text{ or } u > u_{\text{max}} 
\end{cases}
\]

Notice that in general that the partition number $k \ll n$ (number of observations), and due to the nature of partition for $Vu \in \mathbb{R}$ (discourse)

\[
\sum_{j=1}^{k} \mu_{E_j}(u) = 1.
\]
7.3 Markov Chain with Fuzzy states

Accurately speaking, the model investigated here is a Markov chain \( \{X_n, n = 1, 2, \ldots \} \) having fuzzy states, not a fuzzy Markov chain \( \{\tilde{X}_n, n = 1, 2, \ldots \} \). The difference between them is that for any given time \( n = n_0 \), in the former case, \( X_{n_0} \) is a random variable \( X_{n_0}(\omega) \), but in the latter case, for a given time \( n_0 \), \( \tilde{X}_{n_0}(\omega) \) is a fuzzy random variable. The overlap between them is that both have fuzzy subsets \( \{\tilde{E}_1, \tilde{E}_2, \tilde{E}_3\} \) as their values.

Assuming the discourse of the log-return is \( \mathbb{R} = (-\infty, \infty) \), the fuzzy partition on \( \mathbb{R} \) is \( P = \{\tilde{E}_1, \tilde{E}_2, \tilde{E}_3\} \). The Markov chain is denoted as \( \{X_n, n = 1, 2, \ldots \} \) and the observation vector of log-return is \( \{u_1, u_2, \ldots, u_{2000}\} \). Notice that, the time \( n \) for the ZAR/USD five-minute observations is only piecewise equally-spaced. However, as an attempt of approximation we still treat the whole sequence as continuously updated records.

7.3.1 Markov Chains

Let \( \{X_n, n = 1, 2, \ldots \} \) be a Markov Chain with a state-space \( \mathbb{S} = \{0, 1, \ldots \} \).

**Definition 12** \( \{X_n, n = 1, 2, \ldots \} \) is called a Markov Chain if

\[
\Pr[X_{n+1} = j | X_0 = i_0, \ldots, X_{n-1} = i_{n-1}, X_n = i] = \Pr[X_n = j | X_{n-1} = i] \triangleq p_{ij}
\]

where \( i, j, i_0, \ldots, i_{n-1} \in \mathbb{S} \). If \( \mathbb{S} \) is finite, \( \{X_n, n = 1, 2, \ldots \} \) is called a finite Markov Chain. 

\( p_{ij} = \Pr[X_{n+1} = j | X_n = i] \) is called one-step transition probability and the matrix \( P \triangleq (p_{ij}) \) is called the stochastic (or transition probability) matrix.

It is a well-known fact that if the initial distribution \( \rho_0 \) and the transition probability matrix are known, then the Markov chain is fully determinable since its joint distribution (at any time) can be fully expressed in terms of the initial distribution and the transition probabilities. Furthermore, the maximum likelihood estimator of the initial probability distribution and transition probabilities are well developed.
7.3.2 Likelihood theory for Markov chains

The parameter of interest is a point in the space of all transition matrices. Let

$$N_{ij}(n) = \sum_{l=1}^{n} (X_{l-1} = i, X_{l} = j)$$

sum the number of $i,j$-transitions. If $N_{ij}(n) = n_{ij}$ the likelihood takes the form

$$L(p_0, P) = p_{X_0X_1} \times p_{X_1X_2} \times \cdots \times p_{X_{N-1}X_N}.$$

Let

$$L_i(P) = \prod_{j \in S} P_{ij}^{n_{ij}}$$

which depends only on the elements in the $i$th row $P_{ij}$ of $P$. $S$ is the state space (assuming it contains finite elements). The likelihood can be written as

$$L(p_0, P) = p_0(x_0) \prod_{i \in S} L_i(P).$$

In other words, we are estimating $|S|$ independent probability functions.

From the log-likelihood function

$$l(p_0, P) = l(p_0) + \sum_{i \in S} l_i(P_{ij}).$$

We want to maximize $l$ subject to the constraints

$$p_0 \cdot 1 = 1,$$
$$P \cdot 1 = 1.$$

Each of these maximizations can be done separately using Lagrange multipliers by differentiat-
ing a term of the form

\[ t_i(P_i) + \lambda(P_i, \mathbf{1} - 1) \]

\[ = \sum_{j \in S} n_{ij} \log p_{ij} + \lambda(\sum_{j \in S} p_{ij} - 1). \]

Setting the derivatives equal to zero and writing

\[ n_i = \sum_{j \in S} n_{ij}, \]

we obtain

\[ \hat{p}_{ij} = \frac{n_{ij}}{n_i}, n_i > 0 \]

and

\[ \bar{p}_0(t) = \frac{\sum d(i = x_0)}{n - 1}. \]

Conditional on \( n_{ij}, \hat{p}_{ij} \) is asymptotically normal, i.e.,

\[ \sqrt{n_i}(\hat{p}_{ij} - p_{ij}) \xrightarrow{d} N(0, \hat{p}_{ij}(1 - \hat{p}_{ij})). \]

Therefore, the asymptotic confidence interval can be

\[ \left[ \hat{p}_{ij} - \frac{z_{1- \alpha / 2} \sqrt{\hat{p}_{ij}(1 - \hat{p}_{ij})}}{\sqrt{n_i}}, \hat{p}_{ij} + \frac{z_{1- \alpha / 2} \sqrt{\hat{p}_{ij}(1 - \hat{p}_{ij})}}{\sqrt{n_i}} \right] \]

where the asymptotic standard error is

\[ \sqrt{\frac{\hat{p}_{ij}(1 - \hat{p}_{ij})}{n_i}}. \]

An approximate \( 100(1 - \alpha)\% \) confidence interval for \( p_{ij} \) given \( n_i \) is

\[ \left[ \sin \left( \sqrt{\arcsin(p_{ij})} - \frac{z_{1- \alpha / 2}}{2n_i} \right), \sin \left( \sqrt{\arcsin(p_{ij})} + \frac{z_{1- \alpha / 2}}{2n_i} \right) \right] \]
7.3.3 Likelihood estimator for a Markov Chain with Fuzzy States

For a Markov chain with fuzzy state space $P = \{\tilde{E}_1, \tilde{E}_2, \ldots, \tilde{E}_k\}$ the initial probability distribution is

$$p_0^j = \Pr [\tilde{E}_j], \; j = 1, 2, \ldots, k$$

and the transition probabilities are

$$p_{ij}^k = \Pr \left[ X_{n-1} \in \tilde{E}_j \mid X_n \in \tilde{E}_i \right], \; j = 1, 2, \ldots, k, \; n = 1, 2, \ldots.$$

Denote $n_{ij}^k$ as the sum of observations $\{u_1, u_2, \ldots, u_n\}$ falling in fuzzy subset $\tilde{E}_i \; (i = 1, 2, \ldots, k)$. Notice that the definition of the sum of observations $\{u_1, u_2, \ldots, u_n\}$ falling in fuzzy subset $\tilde{E}_i$ $(i = 1, 2, \ldots, k)$ is very similar to that of classical set theory (i.e., the partition is clear sets, say, $P = \{E_1, E_2, \ldots, E_k\}$, where $E_1 \cup E_2 \cup \cdots \cup E_k = \mathbb{R}$ and $E_i \cap E_j = \emptyset, \; i \neq j$), the number falling in set $E_j$ is simply

$$n_i = \sum_{l=1}^{n-1} \psi_{E_l}(u_l), \; i = 1, 2, \ldots, k.$$

Replacing the indicator function $\psi_{E_l}(u_l)$ by the membership function, then $n_{ij}^k$ is defined as

$$n_{ij}^k = \sum_{l=1}^{n-1} \mu_{E_l}(u_l) \psi_{E_l}(u_{l+1}), \; i, j = 1, 2, \ldots, k,$$

For the number from state $i$ (at time $t$) transits into $j$ (at time $t+1$), $n_{ij}$, in classical set theory the case is simply

$$n_{ij} = \sum_{l=1}^{n-1} \psi_{E_l}(u_l) \psi_{E_j}(u_{l+1}), \; i, j = 1, 2, \ldots, k,$$

and therefore in the fuzzy state case, the number from state $i$ (at time $t$) transits into $j$ (at time $t+1$), $n_{ij}^k$,

$$n_{ij}^k = \sum_{l=1}^{n-1} \mu_{E_l}(u_l) \mu_{E_j}(u_{l+1}), \; i, j = 1, 2, \ldots, k.$$

Then the maximum likelihood estimators for initial distributional probabilities are

$$\hat{p}_0^i = \frac{n_{ij}^k}{n-1}, \; i = 1, 2, \ldots, k.$$
And the maximum likelihood estimators for one-step transition probabilities are

\[ \hat{\rho}_{ij} = \frac{n_{ij}^e}{n_i^e}, \ i, j = 1, 2, \ldots, k. \]

Therefore the estimated one-step transition probability matrix \( \hat{P}^e = (\hat{\rho}_{ij})_{k \times k} \) can be used in the forecasting exercise. The following proposition is a matrix version of S. Q. Chen and C. Z. Guo (1994).

**Proposition 13** The maximum likelihood estimator of the transition probability matrix \( \hat{P}^e = (\hat{\rho}_{ij})_{k \times k} \) satisfies the property

\[ \hat{P}^e_{k \times 1} \cdot 1_{k \times 1} = 1_{k \times 1} \]

where \( 1_{k \times 1} \) is a column vector of \( k \) entries of 1.

**Proof.** At the \( i \)-th element of column vector \( \hat{P}^e_{k \times 1} \cdot 1_{k \times 1} \), in terms of matrix multiplication the rule is \( \sum_{j=1}^{k} \hat{\rho}_{ij} \). Note that

\[
\sum_{j=1}^{k} \hat{\rho}_{ij} = \sum_{j=1}^{k} \left( \frac{n_{ij}^e}{n_i^e} \right) = \frac{1}{n_i^e} \sum_{j=1}^{k} n_{ij}^e = \frac{1}{n_i^e} \sum_{j=1}^{k} \left( \sum_{l=1}^{n-1} \mu_{\hat{E}_l}(u_l) \mu_{\hat{P}_{ij}}(u_{l-1}) \right) = \frac{1}{n_i^e} \sum_{l=1}^{n-1} \mu_{\hat{E}_l}(u_l) \left( \sum_{j=1}^{k} \mu_{\hat{P}_j}(u_{l+1}) \right) = \frac{1}{n_i^e} \sum_{l=1}^{n-1} \mu_{\hat{E}_l}(u_l) \times 1 = 1
\]"
by recalling that for a fuzzy partition, \( \sum_{j=1}^{n} \mu_{E_j}(u_i) = 1. \)

### 7.4 Prediction using Markov chain theory

In traditional Markov chain theory, if the Markov chain \( \{X_n\} \) is currently in the state \( E_i \), then the most likely state at time \( n \), should be determined by the transition probabilities \( \{p_{11}, p_{12}, \ldots, p_{nk}\} \). If there is a state \( j_0 \) such that

\[
\pi_{j_0} = \max_{j \in \{1, 2, \ldots, k\}} \{p_{j_0 j_1}, \ldots, p_{j_0 j_k}\}
\]

then the state \( E_{j_0} \) is the most likely state the Markov chain \( \{X_n\} \) would be occupying at time \( n \).

Similarly, in the case of a Markov chain with fuzzy states, the forecasting problem is not as direct as in the case of classical Markov chains, because here it deals with two kind of uncertainties, randomness and fuzziness simultaneously.

Given the observation of a Markov chain with fuzzy states \( \{X_n\}, u_m \), at time \( m \), denote \( \mu_m \) as a row vector of the membership degree of observation \( u_m \) with respect to all the fuzzy states, \( \{E_1, E_2, \ldots, E_k\} \), i.e.,

\[
\mu_m = \left( \mu^E_{E_1}(u_m), \mu^E_{E_2}(u_m), \ldots, \mu^E_{E_k}(u_m) \right)
\]

which may be understood as an overall index showing the degree of observation \( u_m \) (at time \( m \)) subordinating to fuzzy subsets \( E_1, E_2, \ldots, E_k \), therefore indicating the state occupation possibilities due to the fuzziness. Then at time \( m + 1 \), it is necessary to predict the state occupation possibilities first, i.e., calculate the state occupation possibility vector \( \mu_{m+1} \)

\[
\mu_{m+1} = \mu_m \cdot \left( \hat{P}^{x}_{k\times k} \right)^T.
\]

Therefore, in terms of the estimator of \( \hat{P}^{x}_{k\times k} \), the estimated state occupation possibility vector \( \hat{\mu}_{m+1} \)

\[
\hat{\mu}_{m+1} = \mu_m \cdot \left( \hat{\hat{P}}^{x}_{k\times k} \right)^T.
\]
where
\[
\hat{\mu}_{m+1} = \left( \hat{\mu}_{E_1}(u_{m+1}), \hat{\mu}_{E_2}(u_{m-1}), \ldots, \hat{\mu}_{E_k}(u_{m+1}) \right).
\]

Notice that
\[
P_{k \times k} = \begin{bmatrix}
  p_{11} & p_{12} & \cdots & p_{1k} \\
  p_{21} & p_{22} & \cdots & p_{2k} \\
  \vdots & \vdots & \ddots & \vdots \\
  p_{k1} & p_{k2} & \cdots & p_{kk}
\end{bmatrix}
\]

and
\[
\mu_m \cdot (P_{k \times k})' = \left( \mu_{E_1}(u_m), \mu_{E_2}(u_m), \ldots, \mu_{E_k}(u_m) \right)
\]

Therefore,
\[
\hat{\mu}_{E_j}(u_{m+1}) = \sum_{i=1}^{k} \mu_{E_i}(u_m) p_{ij}, \quad j = 1, 2, \ldots, k.
\]

The maximal membership principle is used to determine which fuzzy state the Markov chain could possibly occupy, i.e., there exists a fuzzy state \(E_j\) such that
\[
\hat{\mu}_{E_j}(u_m) = \max_{j_0 \in \{1, 2, \ldots, k\}} \left\{ \hat{\mu}_{E_1}(u_{m-1}), \hat{\mu}_{E_2}(u_{m+1}), \ldots, \hat{\mu}_{E_k}(u_{m+1}) \right\}.
\]

### 7.5 The Estimator for the Observation \(u_{m-1}\)

To obtain an estimate of \(u_{m-1}\), a defuzzification may be necessary. To transform fuzzy results into crisp ones, a defuzzification is performed. Defuzzification is defined as a mapping from fuzzy sets \(\tilde{A}\) to elements of the universe considered significant with respect to \(A\)
\[
\mathcal{F}^{-1} : \mathcal{F}(U) \rightarrow U.
\]
Equivalently, representation defuzzification is a mapping of membership functions (characterising fuzzy sets) to significant elements

\[ F^{-1}_\mu : \{ \mu_{\tilde{A}}(u) | \tilde{A} \in F(U), u \in U \} \rightarrow U. \]

The standard defuzzification method calculates the centroid of the centre of gravity (COG) of the area under the membership function

\[ F^{-1}_{COG}(\tilde{A}) = \frac{\int_U \mu_{\tilde{A}}(u) \, u \, du}{\int_U \mu_{\tilde{A}}(u) \, du}. \]

A modified centroid of largest area method uses only the largest sub-area under the membership function.

The centroid defuzzification has the following properties: Consistency, section invariance, monotonicity, linearity, and scale invariance. The output range is not completely used and there are no forbidden zones. The characteristic is steady and monotonous. The amplification is relatively low for medium firing grades.

Therefore, in terms of the defuzzification method, if the fuzzy state is \( \tilde{E}_{m} \) a point estimator for \( u_{m+1} \) is

\[ \hat{u}_{m-1} = \frac{\int_U \mu_{\tilde{E}_{m}}(u) \, u \, du}{\int_U \mu_{\tilde{E}_{m}}(u) \, du}. \]

**Remark 14** Let \( y_1, y_2, \ldots, y_k \) be the kernels of \( \tilde{E}_1, \tilde{E}_2, \ldots, \tilde{E}_k \) respectively, then

\[ \hat{u}_{m+1} = \frac{\sum_{j=1}^{k} \mu_{\tilde{E}_j}(u_{m-1}) y_j}{\sum_{j=1}^{k} \mu_{\tilde{E}_j}(u_{m-1})} \]

is an alternative estimator for \( u_{m+1} \). The kernel of a fuzzy subset

\[ \text{ker}(\tilde{E}_j) = \min \{ u \in \mathbb{R} : \mu_{\tilde{E}_j}(u) = 1 \}. \]
7.5.1 Point Estimate of FX rate and its Confidence Interval

The log-return of FX rate at time \( t + 1 \), denoted as \( u_{t+1} \) is

\[
u_{t+1} = \ln \left( \frac{r_{t+1}}{r_t} \right), \quad t \geq 0
\]

where \( r_t, r_{t-1} \) are the exchange rates at time \( t \) and \( t-1 \) respectively.

If the estimate of \( u_{t+1} \), \( \hat{u}_{t+1} \) is available, then let

\[
\hat{u}_{t+1} = \ln \left( \frac{\hat{r}_{t+1}}{r_t} \right), \quad t \geq 0
\]

because the FX rate \( r_t \) is available at time \( t \).

Then the estimate of \( \hat{r}_{t+1} \),

\[
\hat{r}_{t+1} = r_t \times \exp (\hat{u}_{t+1}).
\]

If we have calculated the confidence interval for \( \hat{u}_{t+1} \), \( 1 - \alpha = 0.95 \), denoted as \( [u^-_{t+1}, u^+_{t+1}] \), then the confidence interval for \( \hat{r}_{t+1} \) should be \( [r_t \exp (u^-_{t+1}), r_t \exp (u^+_{t+1})] \).

7.6 Fuzzy Markov Chain Theory applied to ZAR/USD 5 Minute Observations

7.6.1 Overall ZAR/USD 5 Minute Data Set (01/07/2002 until 31/01/2003)

In Section 2 we defined the fuzzy partition \( \mathbb{P} = \{ E_1, E_2, E_3 \} \) with membership functions \( \mu_{E_1}(u), \mu_{E_2}(u), \mu_{E_3}(u) \). In this section we investigate the Markov chain having fuzzy states. First we calculate the transition probability matrices \( \hat{P}^{(k)} = \left[ \hat{p}^{(k)}_{ij} \right] \), \( k = 1, 2, \ldots, 20 \). We then aim to predict the state occupation possibilities at time \( m+k \), i.e., the estimated state occupation possibility vector \( \hat{\mu}_{m+k} \). The maximal membership principle, mentioned earlier, can be used to determine which fuzzy state the Markov chain can possibly occupy. These results are then compared with the log returns partitioned by the membership functions into the fuzzy states. If the state prediction is correct at least 60% of the time, then we accept the state predictability. Finally, we wish to predict the foreign exchange rate in the short term. We estimate the forex rate \( \hat{r}_{m+k} \) at time \( m \) and then compare the estimated prediction to the
actual forex rate observed. These calculations are performed in Microsoft Excel. We show some of the intermediate calculations and the final results. The detailed calculations can be found in Appendix B. The estimated one-step transition probability matrix, for the overall data set, is

\[
\hat{P}^{(1)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47058 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.26411
\end{bmatrix}
\]

where the asymptotic standard error for this transition probability matrix is

\[
SE = \begin{bmatrix}
0.00518 & 0.00500 & 0.00579 \\
0.00622 & 0.00624 & 0.00619 \\
0.00581 & 0.00514 & 0.00513
\end{bmatrix}
\]

The stationary transition probability matrix is

\[
P = \begin{bmatrix}
0.36133 & 0.27898 & 0.35800 \\
0.36134 & 0.27899 & 0.35801 \\
0.36129 & 0.27895 & 0.35796
\end{bmatrix}
\]

Then it is expected that an estimated k-step transition probability matrix is

\[
\hat{P}^{(k)} = \left(\hat{P}^{(1)}\right)^k, \ k = 1, 2, \ldots, 20.
\]

**Remark 15** It is definitely feasible for an estimator of \(\hat{P}^{(k)}\) to be directly calculated from the original data, with better accuracy. However, with the Markovian assumption of the FX data, using the k-power method (above) should be adequate.

Using the k-power method yields
\[
\begin{align*}
\mathbf{P}_{3 \times 3}^{(2)} &= \\
&= \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.36993 & 0.25590 & 0.26111 \\
\end{bmatrix} \\
&= \begin{bmatrix}
0.33802 & 0.27870 & 0.33828 \\
0.35932 & 0.28497 & 0.35661 \\
0.34279 & 0.27649 & 0.38072 \\
\end{bmatrix}
\end{align*}
\]

Continuing in this manner until \( k = 20 \), to obtain

\[
\begin{align*}
\mathbf{P}_{3 \times 3}^{(20)} &= \\
&= \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.36993 & 0.25590 & 0.26111 \\
\end{bmatrix}^{20} \\
&= \begin{bmatrix}
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
\end{bmatrix}
\end{align*}
\]

Appendix B contains all 20 transition probability matrices. We also show the intermediate calculations used to obtain \( \mathbf{P}_{3 \times 3}^{(1)}, \mathbf{P}_{3 \times 3}^{(2)}, \mathbf{P}_{3 \times 3}^{(3)} \) from the original data.

The prediction of the state is

\[
\hat{\mu}_{m,k} = \mu_m \cdot (\mathbf{P}_{3 \times 3}^{(k)})^T, \quad k = 1, 2, \ldots, 20
\]

where

\[
\begin{align*}
\mu_m &= (\mu_{\hat{E}_1}(u_m), \mu_{\hat{E}_2}(u_m), \mu_{\hat{E}_3}(u_m)) \\
\hat{\mu}_{m,k} &= (\hat{\mu}_{\hat{E}_1}(\hat{u}_{m+k}), \hat{\mu}_{\hat{E}_2}(\hat{u}_{m+k}), \hat{\mu}_{\hat{E}_3}(\hat{u}_{m+k})) \\
k &= 1, 2, \ldots, 20
\end{align*}
\]

As to which state the \( k \)-step prediction return \( \hat{\mu}_{m,k} \) would occupy, we use the maximum

195
membership principle to determine for each $k$:

$$
\hat{\mu}_{E_k}(\hat{u}_{m+k}) = \max \left\{ \hat{\mu}_{E_1}(\hat{u}_{m+k}), \hat{\mu}_{E_2}(\hat{u}_{m+k}), \hat{\mu}_{E_3}(\hat{u}_{m+k}) \right\}.
$$

Since $\mu_m = (0.59413, 0.40587, 0)$, then

$$
\hat{\mu}_{m-1} = \begin{bmatrix}
0.59443 & 0.40557 & 0
\end{bmatrix}^T
\begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.48999 & 0.23590 & 0.26411
\end{bmatrix}
\begin{bmatrix}
0.26433 & 0.33621 & 0.38722
\end{bmatrix}
$$

Continuing in this manner, we note that $\hat{\mu}_{m+k}$ until $\hat{\mu}_{m+20}$ is calculated and the result is shown in Appendix B.

As to which state the $k$-step prediction return $\hat{\mu}_{m-k}$ would occupy, we use the maximum membership principle to determine for each $k$:

$$
\hat{\mu}_{E_k}(\hat{u}_{m+k}) = \max \left\{ \hat{\mu}_{E_1}(\hat{u}_{m+k}), \hat{\mu}_{E_2}(\hat{u}_{m+k}), \hat{\mu}_{E_3}(\hat{u}_{m+k}) \right\}.
$$

Since $\hat{\mu}_{m-8}$ until $\hat{\mu}_{m-20}$ is calculated, we therefore can only predict up to 7 steps, i.e., 7 predictions for roughly the next half hour.

We then use the maximum membership principle to determine for each $k$ which state the $k$-step prediction return $\hat{\mu}_{m+k}$ occupies, and compare it with the log returns partitioned by the membership functions into the fuzzy states.
Figure 7.1: Comparison of the Actual versus Predicted Membership State for the overall 5 min ZAR/USD time series

We note that our prediction is true 42.88% of the time, since 3 out of 7 times our prediction is correct. This low hit rate is not desirable, as we would want it to be at least 60%. However, the results may have been different if we could have predicted up to 20 time steps instead of just 7.

We now proceed to the short-term prediction of the exchange rate. Since we observe the ZAR/USD forex rate every 5 minutes, we predict the next 7 observations which covers roughly the next half hour (35 minutes). We have shown that the estimate of \( \hat{r}_{t+1} \) is just, \( \hat{r}_{t+1} = r_t \times \exp(\hat{\mu}_{t-1}) \) where \( \hat{\mu}_{t-1} \) is a point estimator for \( u_{t+1} \) if its fuzzy state is \( \hat{E}_j \). Recall an alternative estimator for \( u_{t_0+k} \) is \( \hat{x}_{t_0+k} = \sum_{j=1}^{3} \hat{\mu}_{E_j}(u_{t_0+k})y_{j} / \sum_{j=1}^{3} \hat{\mu}_{E_j}(u_{t_0+k}) \). For the overall ZAR/USD 5 minute time series the kernels are \( \{y_1, y_2, y_3\} = \{-0.614169, 0, 0.000569\} \). We show the final results below, the calculation is detailed in Appendix B.

The table below illustrates the estimates or predictions \( \hat{r}_{t+k} \) (for \( k = 1, 2, ..., 7 \)), where \( r_t = 8.5645 \), versus the actual forex rate observed. The forecast or prediction error is defined as

\[
\frac{\text{Predicted Forex Rate} - \text{Observed Forex Rate}}{\text{Observed Forex Rate}}
\]
From the above results we note that:

1. The predicted forex rates are consistently lesser than the observed forex rates, i.e., the prediction error is always negative. The prediction errors are negative as a result of the kernel, i.e., \( y_1 = -0.014109 \) is a large negative return.

2. The first prediction's error is relatively small, i.e., -0.39% and therefore acceptable.

3. The prediction errors increase in magnitude as we predict up to 7 time steps. The prediction error increases because we use the previous prediction to estimate the next prediction. Prediction errors of greater than 1% are not desirable.

As mentioned above, the kernel of the fuzzy subset has a great influence on the predicted forex rate. Recall that the kernel of a fuzzy subset \( \text{ker} \left( \mathcal{E}_i \right) = \min \left\{ u \in \mathbb{R} | \mu_{\mathcal{E}_i}(u) = 1 \right\} \). If we experiment and change the kernel of a fuzzy subset \( \text{ker} \left( \mathcal{E}_i \right) = \text{average} \left\{ u \in \mathbb{R} | \mu_{\mathcal{E}_i}(u) = 1 \right\} \), then \( \{y_1, y_2, y_3\} = \{-0.001448, 0, 0.001425\} \). We compute the predictions using the new kernel, the results are shown in the table below:

<table>
<thead>
<tr>
<th>Observed Forex Rate</th>
<th>Predicted Forex Rate</th>
<th>Prediction Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.56750</td>
<td>8.53399</td>
<td>-0.39%</td>
</tr>
<tr>
<td>8.56950</td>
<td>8.49364</td>
<td>-1.06%</td>
</tr>
<tr>
<td>8.56435</td>
<td>8.46574</td>
<td>-1.27%</td>
</tr>
<tr>
<td>8.58250</td>
<td>8.41743</td>
<td>-1.92%</td>
</tr>
<tr>
<td>8.58750</td>
<td>8.37637</td>
<td>-2.42%</td>
</tr>
<tr>
<td>8.58350</td>
<td>8.34147</td>
<td>-2.88%</td>
</tr>
<tr>
<td>8.59150</td>
<td>8.30374</td>
<td>-3.35%</td>
</tr>
</tbody>
</table>

Figure 7.2: Comparison of the Observed versus Predicted Forex Rate for the overall 5 min ZAR/USD time series
Figure 7-3: Comparison of the Observed versus Predicted Forex Rate for the overall 5 min ZAR/USD time series (Modified Kernel)

The results clearly illustrate the impact of the kernel of the fuzzy subsets in predicting forex rates. The original kernel contained an extreme minimum, which resulted in the large negative prediction errors. When using average returns for the kernel, the prediction errors are negligible, i.e., smaller than or equal to 0.3% in absolute values.

Thus far, the fuzzy Markov chain analysis was performed on the entire ZAR/USD 5 minute data set. We now partition the data into two parts. Recall that the initial data set is from 01/07/2002 until 31/01/2003. The two new data sets are from 01/07/2002 until 21/08/2002 and 22/08/2002 until 31/01/2003 respectively. From 01/07/2002 until 21/08/2002 the Rand weakened against the Dollar, this general downward trend caused the Rand to fall from 10.3046 to 10.955 against the Dollar. The latter interval illustrated the strengthening of the Rand against the Dollar, i.e., from 10.885 on the 22/08/2002 to 8.5812 on the 31/01/2003. We now perform the same analysis on the partitioned ZAR/USD 5 minute data sets.

### 7.6.2 Partitioned ZAR/USD 5 minute data set

**First Partition (01/07/2002 until 21/08/2002)**

Simple analysis of the partitioned data set yields,
with the fuzzy partition \( P = \{ F_1, F_2, F_3 \} \) and membership functions \( \mu_{E_1}(u), \mu_{E_2}(u), \mu_{E_3}(u) \)
previously defined.

The estimated one-step transition probability matrix is

\[
\hat{P}^{(1)}_{3 \times 3} =
\begin{bmatrix}
0.28762 & 0.24396 & 0.46802 \\
0.36569 & 0.32386 & 0.31046 \\
0.49973 & 0.26393 & 0.23831
\end{bmatrix}
\]

where the asymptotic standard error for this transition probability matrix is

\[
SE =
\begin{bmatrix}
0.01024 & 0.00971 & 0.01129 \\
0.01265 & 0.01229 & 0.01215 \\
0.01137 & 0.01008 & 0.01033
\end{bmatrix}
\]

The stationary transition probability matrix is

\[
P =
\begin{bmatrix}
0.36517 & 0.27117 & 0.35779 \\
0.36517 & 0.27120 & 0.35782 \\
0.36501 & 0.27105 & 0.35793
\end{bmatrix}
\]

The \( k \)-power method is used to estimate all 20 transition probability matrices, which are included in Appendix B. Also shown is the intermediate calculations used to obtain \( \hat{P}^{(1)}_{3 \times 3} \) from the original data.
The prediction of the state is

$$\tilde{\mu}_{m|k} = \mu_m \cdot \left( \hat{P}_{3\times3}^{(k)} \right)^T, \ k = 1, 2, \ldots, 20$$

where

$$\mu_m = \left( \mu_{E_1}(u_m), \mu_{E_2}(u_m), \mu_{E_3}(u_m) \right)$$

$$\tilde{\mu}_{m,k} = \left( \tilde{\mu}_{E_1}(\tilde{u}_{m-k}), \tilde{\mu}_{E_2}(\tilde{u}_{m-k}), \tilde{\mu}_{E_3}(\tilde{u}_{m-k}) \right)$$

$$k = 1, 2, \ldots, 20$$

As to which state the $k$-step prediction return $\tilde{u}_{m+k}$ would occupy, we use the maximum membership principle to determine for each $k$:

$$\tilde{\mu}_{E_j}(\tilde{u}_{m+k}) = \max \left\{ \tilde{\mu}_{E_1}(\tilde{u}_{m+k}), \tilde{\mu}_{E_2}(\tilde{u}_{m+k}), \tilde{\mu}_{E_3}(\tilde{u}_{m+k}) \right\}.$$  

Since $\mu_m = (0, 0, 1)$, then

$$\tilde{\mu}_{m+1} = \left[ \begin{array}{c} 0.28762 \\ 0.36569 \\ 0.44973 \end{array} \right]$$

Until

$$\tilde{\mu}_{m-20} = \left[ \begin{array}{c} 0.36600 \\ 0.36603 \\ 0.36584 \end{array} \right]$$

$\tilde{\mu}_{m+k}$ for $k = 1, 2, \ldots, 20$ is calculated and the result is shown in Appendix B.

We then use the maximum membership principle to determine for each $k$ which state the $k$-step prediction return $\tilde{u}_{m+k}$ occupies and compare it with the log returns partitioned by the membership functions into the fuzzy states.
<table>
<thead>
<tr>
<th>Actual State Membership</th>
<th>Predicted State Membership</th>
<th>True False</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 7.1: Comparison of the Actual versus Predicted Membership State for the ZAR/USD 1st partition time series

For the first partition of the ZAR/USD data, the state predictions are not acceptable. Only 4 out of 20 times (20%) were the predictions correct.

Next, we estimate the forex rate $r_{m+k}$ at time $m$ and then compare the estimated prediction to the actual forex rate observed. The kernels are now $\{y_1, y_2, y_3\} = \{-0.014169, 0, 0.000569\}$. We show the final results below, the calculation is detailed in Appendix B.

The table below illustrates the estimates or predictions $\hat{r}_{m+k}$ (for $k = 1, 2, ..., 20$), where $r_0 = 10.0215$, versus the actual forex rate.
Figure 7-5: Comparison of the Observed versus Predicted Forex Rate for the ZAR/USD 1st partition time series

From the table of results above we note the following:

1. The predicted forex rates are always smaller than the observed forex rates, i.e., the prediction error is always negative. The prediction errors are negative as a result of the kernels, i.e., $\gamma_1 = -0.014169$ is a large negative return.

2. The first prediction's error can be accepted as it is -0.04%.

3. The prediction errors increase in magnitude, as we predict up to 20 time steps. The prediction error increases because we use the previous prediction to estimate the next prediction. Prediction errors of greater than 1% are not desirable.

Second Partition (22/08/2002 until 31/01/2003)

Brief analysis of the partitioned data set yields,
with the fuzzy partition \( \mathcal{F} = \{ \tilde{F}_1, \tilde{F}_2, \tilde{F}_3 \} \) and membership functions \( \mu_{\tilde{F}_1}(u), \mu_{\tilde{F}_2}(u), \mu_{\tilde{F}_3}(u) \) essentially as before.

The estimated one-step transition probability matrix is

\[
P^{(1)}_{3 \times 3} = \begin{bmatrix}
0.27284 & 0.24101 & 0.48605 \\
0.33249 & 0.33794 & 0.33457 \\
0.47820 & 0.26372 & 0.25808
\end{bmatrix}
\]

where the asymptotic standard error for this transition probability matrix is

\[
SE = \begin{bmatrix}
0.00693 & 0.00580 & 0.00672 \\
0.00720 & 0.00726 & 0.00724 \\
0.00596 & 0.00595 & 0.00590
\end{bmatrix}
\]

The stationary transition probability matrix is

\[
P = \begin{bmatrix}
0.36196 & 0.27833 & 0.35971 \\
0.36196 & 0.27833 & 0.35971 \\
0.36196 & 0.27833 & 0.35971
\end{bmatrix}
\]

The \( k \)-power method is used to estimate all 20 transition probability matrices, which is included in Appendix B. Also presented are the intermediate calculations used to obtain \( P^{(1)}_{3 \times 3} \) from the original data.
The prediction of the state is

$$\hat{\mu}_{m+k} = \mu_m \cdot (\hat{P}^{(k)}_{3 \times 3})^T, \quad k = 1, 2, \ldots, 20$$

where

$$\mu_m = \left(\mu_{E_1}(u_m), \mu_{E_2}(u_m), \mu_{E_3}(u_m)\right)$$

$$\hat{\mu}_{m+k} = \left(\hat{\mu}_{E_1}(\hat{u}_{m+k}), \hat{\mu}_{E_2}(\hat{u}_{m+k}), \hat{\mu}_{E_3}(\hat{u}_{m+k})\right)$$

$k = 1, 2, \ldots, 20$

As to which state the $k$-step prediction return $\hat{u}_{m+k}$ would occupy, we use the maximum membership principle to determine for each $k$:

$$\hat{\mu}_{E_0}(\hat{u}_{m+k}) = \max \left\{ \hat{\mu}_{E_1}(\hat{u}_{m+k}), \hat{\mu}_{E_2}(\hat{u}_{m+k}), \hat{\mu}_{E_3}(\hat{u}_{m+k}) \right\}.$$

Since $\mu_m = (0, 0, 1)$, then

$$\hat{\mu}_{m+1} = \begin{bmatrix} 0.27294 & 0.24701 & 0.48005 \\ 0.23749 & 0.23704 & 0.23457 \\ 0.47820 & 0.25872 & 0.25808 \end{bmatrix}^T$$

Since $\hat{\mu}_{m+10}$ until $\hat{\mu}_{m+20}$, we therefore can only predict up to 9 steps, i.e., 9 predictions for the next 45 minutes.

These calculations are provided in detail in Appendix B. We determine for each $k$ which state the $k$-step prediction return $\hat{u}_{m+k}$ occupies and compare it with the log returns partitioned by the membership functions into the fuzzy states.
Figure 7-6: Comparison of the Actual versus Predicted Membership State for the ZAR/USD 2nd partition time series

We note that our prediction is true 55.56% of the time, since 5 out of 9 times our prediction is correct. This hit rate is still less than 60%, but better than the hit rate for the first partition and the overall data set. These results may have been improved if we were able to predict up to 20 time steps instead of just 9.

We now calculate the predictions \( \hat{r}_{t+k} \) (for \( k = 1, 2, \ldots, 9 \)), where \( r_t = 8.5875 \) and the kernels are now \( \{y_1, y_2, y_3\} = \{-0.001289, 0.000577\} \). We show the final results below, the calculation is detailed in Appendix B.

The table below illustrates the estimates or predictions \( \hat{r}_{t+k} \) (for \( k = 1, 2, \ldots, 9 \)) versus the actual Forex rate.

<table>
<thead>
<tr>
<th>Observed Forex Rate</th>
<th>Predicted Forex Rate</th>
<th>Prediction Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.58850</td>
<td>8.53776</td>
<td>-0.59%</td>
</tr>
<tr>
<td>8.59150</td>
<td>8.59408</td>
<td>-0.22%</td>
</tr>
<tr>
<td>8.57190</td>
<td>8.46767</td>
<td>-1.22%</td>
</tr>
<tr>
<td>8.58675</td>
<td>8.3187</td>
<td>-2.22%</td>
</tr>
<tr>
<td>8.58550</td>
<td>8.3964</td>
<td>-1.85%</td>
</tr>
<tr>
<td>8.58750</td>
<td>8.3607</td>
<td>-2.18%</td>
</tr>
<tr>
<td>8.59250</td>
<td>8.3253</td>
<td>-1.85%</td>
</tr>
<tr>
<td>8.59315</td>
<td>8.2901</td>
<td>-3.53%</td>
</tr>
<tr>
<td>8.66000</td>
<td>8.2550</td>
<td>-4.01%</td>
</tr>
</tbody>
</table>

Figure 7-7: Comparison of the Observed versus Predicted Forex Rate for the ZAR/USD 2nd partition time series
From the results above we observe the following:

1. The predicted forex rates are once again smaller than the observed forex rates, i.e., the prediction error is always negative. The prediction errors are negative as a result of the kernels, i.e., \( y_1 = -0.013289 \) is a large negative return.

2. The first prediction’s error is relatively small, i.e., -0.59% and therefore acceptable.

3. The prediction errors increase in magnitude, as we predict up to 9 time steps. The prediction error increases because we use the previous prediction for the next prediction. Prediction errors of greater than 1% is not desirable.

4. These results are similar to those obtained for the overall ZAR/USD 5 minute data set. Since this subset of data is approximately 75% of the overall data set, we would expect similar behaviour.

### 7.7 Concluding Remarks

1. We attempted to predict the ZAR/USD 5 minute forex rates for the next 20 step-ahead predictions. Also of importance, is to predict which state includes the predicted return. We estimate the predictions on the overall ZAR/USD 5 minute data set (01/07/2002 until 31/01/2003), and thereafter split the data into 2 subsets and estimate 2 new sets of predictions. During the first period (01/07/2002 until 31/08/2002), the Rand experienced a general weakening trend against the US Dollar, whilst during the latter period (22/08/2002 until 31/01/2003), the Rand strengthened against the US Dollar.

2. For the overall ZAR/USD 5 minute data set (01/07/2002 until 31/01/2003), we predicted returns and their corresponding state occupied up to 7 time steps. Our findings suggest that the state prediction was not satisfactory, i.e., it was correct only 42.88% of the time. The exchange rate predictions are consistently smaller than the observed forex rates, and the error in prediction increases as the time step increases. For the first forex rate prediction, the error is relatively small, i.e., -0.39%.
3. For the data set from 01/07/2002 until 21/08/2002, we predicted returns and their respective state occupation up to 20 time steps. The state prediction was rather poor, i.e., 20%. Once again the forex rates predictions are consistently smaller than the observed forex rates, and the error in prediction increases as the time step increases. The first prediction error, i.e., -0.64% is less than 1%, but not as small as in the case of the overall data set.

4. For the latter data set from 22/08/2002 until 31/01/2003, we predicted returns and their respective state occupation up to 9 time steps. The state prediction proved to be best for this data set, i.e., 55.58%. Similarly the forex rate predictions are consistently smaller than the observed forex rates, and the error in prediction increases as the time step increases. The first prediction error, i.e., -0.50% is less than 1%, but not as small as in the case of the overall data set.

5. The forex rate predictions are always smaller than the observed forex rates, because of the respective kernels. Those kernels always included a rather large negative return which affects the forex rate predictions. When experimenting with slight modifications to the kernels, we observe favourable predictions, which clearly illustrates that highly volatile time series with extreme returns result in less desirable results when using fuzzy Markov chain modelling.

6. Since the error in prediction increases as the time step increases, we can conclude that short-term prediction for the very high frequency ZAR/USD time series is favoured over long-term prediction.
Chapter 8

CONCLUDING REMARKS

In providing concluding discussions one has to bear in mind that the South African foreign exchange market exhibited extreme (almost abnormal) volatility during the period under study. This volatility is apparent throughout the various analysis and modelling techniques utilised in this dissertation. We have achieved the objectives we outlined in Chapter 1. We have empirically modelled and explored the South African Rand/United States Dollar (ZAR/USD) exchange rate returns time series, thereby providing a short-term prediction of the ZAR/USD returns.

- Chapter 3 provided the basis by exploring the foreign exchange data via the traditional statistical techniques. We found that the daily ZAR/USD exchange rate has been extremely volatile during the period under study (02/01/1990 - 28/02/2003), and is still volatile to date. In comparison the daily DEM/USD exchange rate has been rather stable during its period under study (02/01/1990 - 31/12/1998). It is also evident that the intraday ZAR/USD data also shows patterns of high volatility. For the 7 month period under study (01/07/2002 - 31/01/2003) the rand moved from 10.30 to 8.58 to the dollar. The distribution of the daily and intradaily ZAR/USD returns (5, 15, 30, 60 minute observations) is very non-normal indeed, since the distributions have extremely high peaks at the mean and much heavier tails than the corresponding normal distributions. In comparison the daily DEM/USD returns distribution is relatively close to the normal distribution.
• We proceeded to fit a stable distribution to the daily and intraday ZAR/USD returns since stable distributions allow for non-normal distributions, with skewness and heavy tails. The stable distribution fits the daily ZAR/USD time series fairly well. In the case of intraday data, the stable distribution is a satisfactory approximation for the returns distribution. The right and left tails are not very well fitted by the stable distribution indicated by the graphical analysis (p-p plots). For very high frequency data, i.e., 5 minute observations, there is also deviation near the centre of the plot, which is due to the large number of zeroes contained in very high frequency data sets.

Fitting experience shows that if the number of zeroes exceeds 25% the p-p plots demonstrate non-linear patterns. Therefore a mixture of a discrete probability distribution and a stable distribution may offer a realistic fitting. However, the current fitting (p-p plots and stabilised p-p plots) of the distributions are limited by the existing software, STABLE 3.04 FORTRAN written by John P. Nolan. The original set of analyses was discarded since the intraday data sets included 25% - 50% zeroes, and clearly demonstrated non-linear patterns. Deviations of the p-p plots from the reference lines were observed at the ends and centres of the plots. The deviation of the p-p plots from the reference lines were greatly evident at the centre of the plots, thus forming a zig-zag line. We then decided to clean the intraday data sets by removing zero returns during non-trading periods, i.e., those returns occurring from midnight to seven in the morning. However, the ZAR/USD intraday trading is not strictly from seven in the morning to midnight daily, on occasion there is no trading activity from nine or ten in the evening. Thus, these intraday subsets used in the stable analyses presented here may also include a fair amount of zero returns.

• Next we modelled the SA forex market by using a multi-agent neural network approach. Our findings are therefore, that the ZAR/USD foreign exchange market may be modelled using a feed forward neural network modelled on explicit price dynamics. We recommend using between 100 and 200 agents in the agents layer of the neural network and, using the research of Zimmermann, Grothmann and Neuneyer, suggest an average agents database of 7 - 9 input signals. Neural networks as a class of model, for modelling the dynamics of a foreign exchange market have vast potential, the bounds of which have not been tested by the relatively simplified neural network model implemented in this exploratory
analysis. Further analysis using slightly more complex neural network implementations such as error correction neural networks is certainly recommended. In this study we used neural networks to model daily foreign exchange data, so too can neural networks be used to model intraday data. However, due to time constraints we did not apply the neural networks to model intraday data, as only daily data was initially available and intraday data became available at a much later stage.

- The analysis proceeded to examine the suitability of the Kalman filter for analysing the dynamics of the foreign exchange rates. Various state-space models were applied to the different data sets. It was found that the daily DEM/USD and daily ZAR/USD returns are better modelled within the state-space framework using multiple explanatory variables as opposed to being modelled using a single variable. For the daily DEM/USD returns, the state-space model that produces the best estimate is the time varying random coefficients model with random walk errors. For the daily ZAR/USD returns, the state-space model that produces the best estimate is the VAR(1) model without drift. We then recommend, for future research, modelling the ZAR/USD returns using a Kalman filter that is adapted to a non-linear state space model, or a state-space model that has non-normal noise, and to compare these results to the EViews linear models.

- Finally we modelled the very high frequency data (5 minute observations) using fuzzy Markov chain theory. We also attempted to provide some short-term prediction of the ZAR/USD returns, i.e., 20 step-ahead predictions. Also of importance, is to predict which state includes the return. We estimate the predictions on the overall ZAR/USD 5 minute data set (01/07/2002 until 31/01/2003), and thereafter split the data into 2 subsets and estimate 2 new sets of predictions. During the first period (01/07/2002 until 21/08/2002), the Rand experienced a general weakening trend against the US Dollar, whilst during the latter period (22/08/2002 until 31/01/2003), the Rand strengthened against the US Dollar. For all three ZAR/USD 5 minute data sets the state prediction was not really satisfactory, i.e., it was always less than 60%. The exchange rate predictions are consistently smaller than the observed forex rates, and the error in prediction increases as the time step increases. The forex rate predictions are always smaller than the observed
forex rates, because of the respective kernels. These kernels always included a rather large negative return which affects the forex rate predictions. When experimenting with slight modifications to the kernels, we observe favourable predictions, which clearly illustrates that highly volatile time series with extreme returns result in less desirable results when using fuzzy Markov chain modelling. Since the error in prediction increases as the time step increases, we can conclude that short-term prediction for the very high frequency ZAR/USD time series is favoured over long-term prediction.

We have thus achieved the objectives of this Masters dissertation, to empirically model and explore forex rates, in particular, the South African Rand/United States Dollar (ZAR/USD) exchange rate returns time series. We also attempted to provide predictions of the ZAR/USD returns. It seems that for high frequency ZAR/USD time series, the short-term predictions are more desirable than the long-term predictions, which is typical for high frequency time series, especially if the period under study is lengthy and volatile. We must bear in mind that trying to predict the ZAR/USD forex rates during such a volatile period is a formidable task. Our analysis has shown that there are various models and techniques that may be used to model high frequency foreign exchange time series data. This Masters dissertation presents analysis and results of modelling the high frequency ZAR/USD data utilising some of these methods. Since we present a few of these alternative methods, we have only covered the basic models in the form of linear Kalman filters, multi-agent fixed forward neural networks modelled on explicit market price dynamics and fuzzy Markov chain modelling. These models form the basis and foundation of much more complex models, for example, there has been significant advances in the fields of error correction and recurrent neural networks as well as non-linear Kalman filters that are worth investigation.

It would be inappropriate to attempt to directly compare these different methods as each has its merits and demerits when applied to time series models with different characteristics. Our empirical results, do however suggest that further in depth investigation into more complex implementations of these theories is certainly warranted for modelling high frequency foreign exchange data.
Bibliography


35] LeBaron, B. (1998), "Agent Based Computational Finance: Suggested Readings and Early Research", Forthcoming: Journal of Economic Dynamics and Control, Graduate School of International Economics and Finance, Brandeis University, 415 South Street


Internet URL: http://www.cs.unc.edu/~welch/kalman/maybeck.html


Available from Internet URL: http://academic2.american.edu/~jpnolan/stable/mle.ps


219
Appendix A

SENN CODES

A.1 SENN MODEL FILE

APPLICATION ZAR/USD Explicit
MODE DAY WEEK 5
FROM 03.01.1990 TO MAX // We can change MAX to experiment with periods excluding
the extreme volatility in December 2001
TRAINING FROM 03.01.1990 TO 03.09.1995 // We have chosen a training set in the first
half of the training set.
VALIDATION FROM 03.09.1995 TO 03.09.1996 // I have used the same validation set as
used in the original model - we may have to change this at a later stage as well
// VALIDATION RANDOM 0%
// END OF HEADER DECLARATION SECTION
INPUT CLUSTER mlp.input // input cluster is called mlp.input - names left as used by
Zimmermann, Grothmann and Nauncier
BEGIN ZARUSD "ZARAND/USDOLLAR"
// Rand Dollar exchange Rate
x - FILE data/mydata.txt COLUMN 1
INPUT = scale((x - x(-1)) / x(-1)) // Input data transformations
INPUT = scale((x - 2 + x(-1) - x(-2)) / x)
BEGIN USDGBP "USDOllAR/POUND"
    //US DOLLAR POUND exchange Rate
    x = FILE data/mydata.txt COLUMN 2
    INPUT = scale((x - x(-1)) / x(-1))
    INPUT = scale((x - 2 * x(-1) + x(-2)) / x)
END

BEGIN NAME "CHF/DOLLAR"
    //SWISS FRANC - US DOLLAR Marker
    x = FILE data/mydata.txt COLUMN 3
    INPUT = scale((x - x(-1)) / x(-1))
    INPUT = scale((x - 2 * x(-1) + x(-2)) / x)
END

BEGIN DEMUSD "DEMARKER/USDOLLAR"
    //Mark - US DOLLAR exchange Rate
    x = FILE data/mydata.txt COLUMN 4
    INPUT = scale((x - x(-1)) / x(-1))
    INPUT = scale((x - 2 * x(-1) + x(-2)) / x)
END

BEGIN ZARGBP "ZAR GBP RATE"
    //Rand Sterling Exchange Rate
    x = FILE data/mydata.txt COLUMN 5
    INPUT = scale((x - x(-1)) / x(-1))
    INPUT = scale((x - 2 * x(-1) - x(-2)) / x)
END

BEGIN GOLD "GOLD BULLION DOLLAR PRICE"
    //Gold Bullion Dollar Price
    x = FILE data/mydata.txt COLUMN 6

INPUT = scale((x - x(-1)) / x(-1))

INPUT = scale((x - 2 * x(-1) - x(-2)) / x)
END
BEGIN ALSI "JSE ALL SHARE INDEX"
   // JSE ALL SHARE INDEX
   x = FILE data/mydata.txt COLUMN 7
INPUT = scale((x - x(-1)) / x(-1))

INPUT = scale((x - 2 * x(-1) + x(-2)) / x)
END
TARGET CLUSTER mlp.price // target cluster
BEGIN price
   x = FILE data/mydata.txt COLUMN 1
   TARGET = 100 * ln(x(1) / x)
   ASSIGN TO channel
END
TARGET CLUSTER mlp.agents // second target cluster - this is where we will vary the
number of agents in our investigation
BEGIN agents behavior
   x = FILE data/mydata.txt COLUMN 1
   TARGET = 100 * ln(x(1) / x)
   TARGET = 100 * ln(x(1) / x)
   TARGET = 100 * ln(x(1) / x)
   TARGET = 100 * ln(x(1) / x)
   TARGET = 100 * ln(x(1) / x)
   TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

 TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)

TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = \(100 \cdot \ln(x(1) / x)\)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
\[
\begin{align*}
\text{TARGhT} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x) \\
\text{TARGET} &= 100 \times \ln(x(1) / x)
\end{align*}
\]
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)
 TARGET = 100 * ln(x(1) / x)
TARGET = 100 * ln(x(1) / x)

END

SIGNAL //signal definitions will help us to evaluate the models performance

BEGIN hit rate = NORMSUM(signal)
    t = TARGET channel
    o = OUTPUT channel
    SIGNAL = If t * o > 0 Then 1 Else 0
END

BEGIN Rel
y = FILE data/mydata.txt COLUMN 1
o = OUTPUT channel
SIGNAL = (y(1) / y - 1) * sign(o)
END

BEGIN realised potential = Relsum(signal1, signal2)
    y = FILE data/mydata.txt COLUMN 1
    o = OUTPUT channel
    SIGNAL = (y(1) / y - 1) * sign(o)
    SIGNAL = abs(y(1) / y - 1)
BEGIN Backtransformation of forecasts

\[ y = \text{FILE data/mydata.txt COLUMN 1} \]
\[ \sigma = \text{OUTPUT channel} \]
\[ \text{SIGNAL } y(1) \]
\[ \text{SIGNAL} = y \times (1 - \sigma / 100) \]
END

BEGIN Buy & Hold

\[ y = \text{FILE data/mydata.txt COLUMN 1} \]
\[ \text{SIGNAL} = y(1) / y - 1 \]
END

BEGIN Naive Prognosis

\[ y = \text{FILE data/mydata.txt COLUMN 1} \]
\[ \text{SIGNAL} = (y(1) / y - 1) \times \text{sign}(y - y(-1)) \]
END

A.2 SENN TOPOLOGY FILE

```plaintext
class { const nr_price = 1;
  cluster (IN) input;
    cluster (DIM(nr_price),HID) excessDemand;
    cluster (OUT) price;
    cluster (OUT) agents;

  connect (input -> agents);
  connect (agents -> excessDemand);
  connect (excessDemand -> price);
  connect (bias -> agents);
} mlp
```

230
Appendix B

FUZZY MARKOV CHAIN MODELLING CALCULATIONS

B.1 Overall ZAR/USD 5 minute data set (01/07/2002 until 31/01/2003)

Transition Probability Matrices

The estimated one-step transition probability matrix, for the overall ZAR/USD 5 minute data set, is

\[ \hat{P}_{ij} = \frac{n_{ij}^e}{n_i^e}, \text{where } i, j = 1, 2, 3. \]

\[ \hat{P}_{3x3} = \begin{bmatrix} 0.27559 & 0.24783 & 0.47658 \\ 0.33822 & 0.33766 & 0.32712 \\ 0.46996 & 0.26890 & 0.25411 \end{bmatrix} \]

where
\[ n_{ij}^3 = n_{A_{3|4}}^3 = \begin{bmatrix} 2053.28 & 1846.44 & 3550.70 \\ 1927.89 & 1041.25 & 1881.28 \\ 3469.24 & 1962.74 & 1949.48 \end{bmatrix} \]

and

\[ n_i^3 = \begin{bmatrix} 1927.89 & 1041.25 & 1881.28 \end{bmatrix} \]

Since,

\[ n - 1 = 20582 \]

then,

\[ \bar{p}_i = \begin{bmatrix} 0.361987 \\ 0.279425 \\ 0.358637 \end{bmatrix} \]

The estimated two-step transition probability matrix, for the partitioned ZAR/USD 5 minute data set, is

\[ \hat{P}_{3 \times 3}^{(2)} = \begin{bmatrix} 0.36634 & 0.26750 & 0.36610 \\ 0.34958 & 0.30957 & 0.34075 \\ 0.36710 & 0.26775 & 0.36505 \end{bmatrix} \]

where

\[ n_{3 \times 3}^{(2)} = \begin{bmatrix} 2731.01 & 1904.19 & 2729.21 \\ 2010.70 & 1780.53 & 1950.80 \\ 2711.15 & 1973.95 & 2695.36 \end{bmatrix} \]
Since,

\[ n_i \]
\[ n_1 = 7454.86 \]
\[ n_2 = 3751.68 \]
\[ n_3 = 7383.46 \]

and

\[ n - 1 = 20582 \]

Then,

\[ \hat{P} \]
\[ \hat{P}_1 = 0.362203 \]
\[ \hat{P}_2 = 0.279452 \]
\[ \hat{P}_3 = 0.358734 \]

The estimated three-step transition probability matrix, for the overall ZAR/USD 5 minute data set, is

\[ \hat{P}_{3,3}^{(2)} = \begin{bmatrix} 0.35797 & 0.27598 & 0.36305 \\ 0.26106 & 0.27970 & 0.35835 \\ 0.35609 & 0.27955 & 0.35436 \end{bmatrix} \]

where

\[ \hat{n}_{3,3}^{(2)} = \begin{bmatrix} 2727.57 & 2015.02 & 2710.53 \\ 2042.70 & 1723.84 & 1984.58 \\ 2682.29 & 2012.81 & 2088.36 \end{bmatrix} \]

and
If
\[ n^i_1 = 7454.86 \]
\[ n^i_2 = 5751.68 \]
\[ n^i_3 = 7383.46 \]

Since

\[ n - 1 = 20582 \]

Then

\[ \hat{p}^i_1 = 0.302203 \]
\[ \hat{p}^i_2 = 0.279452 \]
\[ \hat{p}^i_3 = 0.358734 \]

Using the $k$-power method yields

\[
\hat{P}_3^{(2)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.16990 & 0.26590 & 0.26411 \\
\end{bmatrix}^2 = \begin{bmatrix}
0.38302 & 0.2787 & 0.33828 \\
0.35932 & 0.28407 & 0.35661 \\
0.34279 & 0.27649 & 0.38072 \\
\end{bmatrix}
\]
\[ \hat{P}_{3 \times 3}^{(3)} = \begin{bmatrix} 0.27559 & 0.24783 & 0.47658 \\ 0.33522 & 0.33766 & 0.32712 \\ 0.46999 & 0.26500 & 0.26411 \end{bmatrix} \]

\[ \hat{P}_{3 \times 3}^{(4)} = \begin{bmatrix} 0.27559 & 0.24783 & 0.47658 \\ 0.33522 & 0.33766 & 0.32712 \\ 0.46999 & 0.26500 & 0.26411 \end{bmatrix} \]

\[ \hat{P}_{3 \times 3}^{(5)} = \begin{bmatrix} 0.27559 & 0.24783 & 0.47658 \\ 0.33522 & 0.33766 & 0.32712 \\ 0.46999 & 0.26500 & 0.26411 \end{bmatrix} \]

\[ \hat{P}_{3 \times 3}^{(6)} = \begin{bmatrix} 0.27559 & 0.24783 & 0.47658 \\ 0.33522 & 0.33766 & 0.32712 \\ 0.46999 & 0.26500 & 0.26411 \end{bmatrix} \]
\[
\tilde{P}_{3 \times 3}^{(7)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.26411 \\
\end{bmatrix}
= \begin{bmatrix}
0.36196 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
\end{bmatrix}
\]

\[
\tilde{P}_{3 \times 3}^{(8)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.26411 \\
\end{bmatrix}
= \begin{bmatrix}
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
\end{bmatrix}
\]

\[
\tilde{P}_{3 \times 3}^{(9)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.26411 \\
\end{bmatrix}
= \begin{bmatrix}
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
\end{bmatrix}
\]
\[
\hat{P}_{3\times 3}^{(10)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.26411 \\
\end{bmatrix}
\]
\[
= \begin{bmatrix}
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
\end{bmatrix}
\]
\[
\hat{P}_{3\times 3}^{(11)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.26411 \\
\end{bmatrix}
\]
\[
= \begin{bmatrix}
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
\end{bmatrix}
\]
\[
\hat{P}_{3\times 3}^{(12)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.26411 \\
\end{bmatrix}
\]
\[
= \begin{bmatrix}
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
\end{bmatrix}
\]
\[
\hat{P}_{3 \times 3}^{(13)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.49999 & 0.26590 & 0.26411 \\
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(14)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.26411 \\
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(15)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.26411 \\
\end{bmatrix}
\]
\[
\hat{P}_{3 \times 3}^{(16)} = \begin{bmatrix}
0.27559 & 0.24788 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.20411 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(17)} = \begin{bmatrix}
0.27559 & 0.24788 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.26411 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(18)} = \begin{bmatrix}
0.27559 & 0.24788 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46999 & 0.26590 & 0.26411 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
\end{bmatrix}
\]
\[
\hat{P}_{3 \times 3}^{(10)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46099 & 0.26500 & 0.26411
\end{bmatrix}

= \begin{bmatrix}
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(20)} = \begin{bmatrix}
0.27559 & 0.24783 & 0.47658 \\
0.33522 & 0.33766 & 0.32712 \\
0.46099 & 0.26500 & 0.26411
\end{bmatrix}

= \begin{bmatrix}
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862 \\
0.36197 & 0.27941 & 0.35862
\end{bmatrix}
\]

**State Prediction**

The prediction of the state is

\[
\hat{x}_{m+k} = \mu_m \cdot \left( \hat{P}_{3 \times 3}^{(k)} \right)^T, \quad k = 1, 2, \ldots, 20
\]

where

\[
\mu_m = \left( \mu_{E_1}(\tilde{u}_m), \mu_{E_2}(\tilde{u}_m), \mu_{E_3}(\tilde{u}_m) \right)
\]

\[
\mu_{m+k} = \left( \mu_{E_1}(\tilde{u}_{m-k}), \mu_{E_2}(\tilde{u}_{m-k}), \mu_{E_3}(\tilde{u}_{m-k}) \right)
\]

\[
k = 1, 2, \ldots, 20
\]

As to which state the k-step prediction return \( \tilde{u}_{m+k} \) would occupy, we use the maximum membership principle to determine for each k:

\[
\tilde{u}_{E_n}(\tilde{u}_{m+k}) = \max \left\{ \tilde{u}_{E_1}(\tilde{u}_{m-k}), \tilde{u}_{E_2}(\tilde{u}_{m-k}), \tilde{u}_{E_3}(\tilde{u}_{m+k}) \right\}
\]
Since \( \mu_m = (0.59443, 0.40557, 0) \), then

\[
\begin{align*}
\hat{\mu}_{m1} &= \begin{bmatrix} 0.59443 & 0.40557 & 0 \end{bmatrix} \begin{bmatrix} 0.27550 & 0.24783 & 0.47658 \\ 0.35522 & 0.33766 & 0.32712 \\ 0.46929 & 0.26590 & 0.26411 \end{bmatrix}^T \\
\hat{\mu}_{m2} &= \begin{bmatrix} 0.59443 & 0.40557 & 0 \end{bmatrix} \begin{bmatrix} 0.36302 & 0.27857 & 0.33828 \\ 0.35932 & 0.28407 & 0.35661 \\ 0.34279 & 0.27549 & 0.38072 \end{bmatrix}^T \\
\hat{\mu}_{m3} &= \begin{bmatrix} 0.59443 & 0.40557 & 0 \end{bmatrix} \begin{bmatrix} 0.35797 & 0.27898 & 0.36305 \\ 0.36185 & 0.27979 & 0.35835 \\ 0.36609 & 0.27955 & 0.35436 \end{bmatrix}^T \\
\hat{\mu}_{m4} &= \begin{bmatrix} 0.59443 & 0.40557 & 0 \end{bmatrix} \begin{bmatrix} 0.36286 & 0.27943 & 0.35775 \\ 0.36194 & 0.27944 & 0.35862 \\ 0.36115 & 0.27935 & 0.35951 \end{bmatrix}^T \\
\hat{\mu}_{m5} &= \begin{bmatrix} 0.59443 & 0.40557 & 0 \end{bmatrix} \begin{bmatrix} 0.36180 & 0.27943 & 0.3588 \\ 0.36197 & 0.27941 & 0.35862 \\ 0.36214 & 0.27942 & 0.35844 \end{bmatrix}^T \\
\hat{\mu}_{m6} &= \begin{bmatrix} 0.59443 & 0.40557 & 0 \end{bmatrix} \begin{bmatrix} 0.362 & 0.27941 & 0.35859 \\ 0.36197 & 0.27941 & 0.35862 \\ 0.36193 & 0.27941 & 0.35866 \end{bmatrix}^T \\
\hat{\mu}_{m7} &= \begin{bmatrix} 0.59443 & 0.40557 & 0 \end{bmatrix} \begin{bmatrix} 0.36196 & 0.27941 & 0.35863 \\ 0.36197 & 0.27941 & 0.35862 \\ 0.36197 & 0.27941 & 0.35862 \end{bmatrix}^T \\
&= \begin{bmatrix} 0.32848 & 0.32849 & 0.32849 \end{bmatrix}^T
\end{align*}
\]
\[ \bar{\mu}_{m+8} = \begin{bmatrix} 0.59443 & 0.40557 & 0 \\ 0.36197 & 0.27941 & 0.35862 \\ 0.36197 & 0.27941 & 0.35862 \end{bmatrix}^T \]

\[ \bar{\mu}_{m+9} = \begin{bmatrix} 0.59443 & 0.40557 & 0 \\ 0.36197 & 0.27941 & 0.35862 \\ 0.36197 & 0.27941 & 0.35862 \end{bmatrix}^T \]

\[ \bar{\mu}_{m+10} = \begin{bmatrix} 0.59443 & 0.40557 & 0 \\ 0.36197 & 0.27941 & 0.35862 \\ 0.36197 & 0.27941 & 0.35862 \end{bmatrix}^T \]

As to which state the \( k \)-step prediction return \( \bar{\mu}_{m+k} \) would occupy, we use the maximum membership principle to determine for each \( k \):

\[ \bar{\mu}_{\mathcal{F}_j} (\bar{\Omega}_{m+k}) = \max \{ \bar{\mu}_{\Sigma} (\bar{\Omega}_{m+k}), \bar{\mu}_{\mathcal{F}_1} (\bar{\Omega}_{m+k}), \bar{\mu}_{\mathcal{F}_2} (\bar{\Omega}_{m+k}) \} \]

\[ \bar{\mu}_{\mathcal{F}_1} (\bar{\Omega}_{m+1}) = \max \begin{bmatrix} 0.26433 & 0.33621 & 0.38722 \end{bmatrix} = 0.38722 \]

\[ \bar{\mu}_{\mathcal{F}_2} (\bar{\Omega}_{m+2}) = \max \begin{bmatrix} 0.34071 & 0.32888 & 0.3359 \end{bmatrix} = 0.34071 \]

\[ \bar{\mu}_{\mathcal{F}_3} (\bar{\Omega}_{m+3}) = \max \begin{bmatrix} 0.32593 & 0.32557 & 0.33099 \end{bmatrix} = 0.33099 \]

\[ \bar{\mu}_{\mathcal{F}_4} (\bar{\Omega}_{m+4}) = \max \begin{bmatrix} 0.3299 & 0.32843 & 0.32797 \end{bmatrix} = 0.3299 \]

\[ \bar{\mu}_{\mathcal{F}_5} (\bar{\Omega}_{m+5}) = \max \begin{bmatrix} 0.32838 & 0.32849 & 0.32859 \end{bmatrix} = 0.3285 \]

\[ \bar{\mu}_{\mathcal{F}_6} (\bar{\Omega}_{m+6}) = \max \begin{bmatrix} 0.3285 \end{bmatrix} = 0.3285 \]

\[ \bar{\mu}_{\mathcal{F}_7} (\bar{\Omega}_{m+7}) = \max \begin{bmatrix} 0.32848 & 0.32849 & 0.32849 \end{bmatrix} = 0.32849 \]

\[ \bar{\mu}_{\mathcal{F}_8} (\bar{\Omega}_{m+8}) = \max \begin{bmatrix} 0.32849 & 0.32849 & 0.32849 \end{bmatrix} = 0.32849 \]

\[ \bar{\mu}_{\mathcal{F}_9} (\bar{\Omega}_{m+9}) = \max \begin{bmatrix} 0.32849 & 0.32849 & 0.32849 \end{bmatrix} = 0.32849 \]

\[ \bar{\mu}_{\mathcal{F}_{10}} (\bar{\Omega}_{m+10}) = \max \begin{bmatrix} 0.32849 & 0.32849 & 0.32849 \end{bmatrix} = 0.32849 \]

\[ \sum_{j=1}^{3} \bar{\mu}_{\mathcal{F}_j} (\bar{\Omega}_{m+k}) \text{ for } k = 1, 2, ..., 10, \text{ i.e.,} \]

242
Return Prediction

Let $y_1, y_2, \ldots, y_k$ be the kernels of $E_1, E_2, \ldots, E_k$ respectively, i.e.,

$$[ -0.014169 \ 0 \ 0.000569 ]$$

$$\sum_{j=1}^{3} \hat{\mu}_{E_j}(u_{m+1})y$$ for $k = 1, 2, \ldots, 10$, i.e.,

Therefore, $\hat{w}_{m+k}$ for $k = 1, 2, \ldots, 10$, i.e.,
Then proceed to compute the predicted forex rates, $\hat{r}_{t+k}$ (for $k = 1, 2, ..., 7$) where $r_t = 8.5645$.

B.2 Partitioned ZAR/USD 5 minute data set

B.2.1 First Partition (01/07/2002 until 21/08/2002)

Transition Probability Matrices

The estimated one-step transition probability matrix, for the partitioned ZAR/USD 5 minute data set, is

$$
\hat{P}_{5 \times 5}^{(1)} = 
\begin{bmatrix}
0.28762 & 0.24336 & 0.16902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581
\end{bmatrix}
$$

where

$$
\hat{\pi}_{3 \times 3}^{(1)} = 
\begin{bmatrix}
561.90 & 475.44 & 916.30 \\
530.25 & 469.60 & 450.17 \\
860.49 & 501.98 & 546.86
\end{bmatrix}
$$

and
\[ n^* \]
\[ n^*_1 = 1953.65 \]
\[ n^*_2 = 1450.02 \]
\[ n^*_3 = 1913.33 \]

Since

\[ n - 1 = 5316 \]

Then

\[ \hat{P}_k \]
\[ \hat{P}_1 = 0.367503 \]
\[ \hat{P}_2 = 0.272760 \]
\[ \hat{P}_3 = 0.359919 \]

The estimated \( k \)-step transition probability matrix for the partitioned ZAR/USD 5 minute data set where \( k = 1, 2, \ldots, 20 \), using the \( k \)-power method yields

\[
\hat{P}_{3 \times 3}^{(2)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.48902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581
\end{bmatrix}
^2
= \begin{bmatrix}
0.33225 & 0.27260 & 0.34435 \\
0.35324 & 0.27582 & 0.36079 \\
0.35441 & 0.27036 & 0.37456
\end{bmatrix}
\]
\[
\hat{P}_{3 \times 3}^{(3)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26303 & 0.28581
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(4)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26303 & 0.28581
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(5)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26303 & 0.28581
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(6)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26303 & 0.28581
\end{bmatrix}
\]
\[
\tilde{P}^{(7)}_{3 \times 3} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581 \\
\end{bmatrix}^{7}
= \begin{bmatrix}
0.36689 & 0.27245 & 0.35948 \\
0.36698 & 0.27248 & 0.35951 \\
0.36673 & 0.27233 & 0.35931 \\
\end{bmatrix}
\]

\[
\tilde{P}^{(8)}_{3 \times 3} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581 \\
\end{bmatrix}^{8}
= \begin{bmatrix}
0.36688 & 0.27244 & 0.35941 \\
0.36686 & 0.27242 & 0.35944 \\
0.36666 & 0.27228 & 0.35925 \\
\end{bmatrix}
\]

\[
\tilde{P}^{(9)}_{3 \times 3} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581 \\
\end{bmatrix}^{9}
= \begin{bmatrix}
0.36676 & 0.27235 & 0.35934 \\
0.36679 & 0.27237 & 0.35937 \\
0.36659 & 0.27223 & 0.35918 \\
\end{bmatrix}
\]
\[
\hat{P}_{3 \times 3}^{(10)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581
\end{bmatrix}^{10}
= \begin{bmatrix}
.36669 & .2723 & .35927 \\
.36672 & .27232 & .3593 \\
.36652 & .27218 & .35911
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(11)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581
\end{bmatrix}^{11}
= \begin{bmatrix}
.36662 & .27225 & .35921 \\
.36665 & .27227 & .35924 \\
.36646 & .27219 & .35904
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(12)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581
\end{bmatrix}^{12}
= \begin{bmatrix}
.36655 & .2722 & .35914 \\
.36658 & .27222 & .35917 \\
.36639 & .27207 & .35898
\end{bmatrix}
\]
\[
\hat{P}_{3\times3}^{(14)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581 \\
\end{bmatrix}^{14}
\]

\[
= \begin{bmatrix}
0.36648 & 0.27214 & 0.35907 \\
0.36651 & 0.27217 & 0.3591 \\
0.36632 & 0.27202 & 0.35891 \\
\end{bmatrix}
\]

\[
\hat{P}_{3\times3}^{(14)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581 \\
\end{bmatrix}^{14}
\]

\[
= \begin{bmatrix}
0.36641 & 0.27209 & 0.359 \\
0.36644 & 0.27212 & 0.35903 \\
0.36625 & 0.27197 & 0.35884 \\
\end{bmatrix}
\]

\[
\hat{P}_{3\times3}^{(15)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581 \\
\end{bmatrix}^{15}
\]

\[
= \begin{bmatrix}
0.36634 & 0.27204 & 0.35894 \\
0.36638 & 0.27207 & 0.35897 \\
0.36618 & 0.27192 & 0.35877 \\
\end{bmatrix}
\]
\[
\hat{P}_{3 \times 3}^{(15)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
.36627 & .27199 & .35887 \\
.36631 & .27201 & .3589 \\
.36611 & .27187 & .35871 \\
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(17)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
.36621 & .27194 & .3588 \\
.36624 & .27198 & .35883 \\
.36604 & .27182 & .35864 \\
\end{bmatrix}
\]

\[
\hat{P}_{3 \times 3}^{(18)} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
.36614 & .27189 & .35873 \\
.36617 & .27191 & .35876 \\
.36597 & .27177 & .35857 \\
\end{bmatrix}
\]

250
\[
\hat{P}^{(19)}_{3 \times 3} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581
\end{bmatrix}^{19} = \begin{bmatrix}
.36907 & .27184 & .35867 \\
.3661 & .27186 & .3587 \\
.3659 & .27172 & .35851
\end{bmatrix}
\]

\[
\hat{P}^{(20)}_{3 \times 3} = \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581
\end{bmatrix}^{20} = \begin{bmatrix}
0.36600 & 0.27179 & 0.3586 \\
0.36603 & 0.27181 & 0.35863 \\
0.36584 & 0.27165 & 0.35844
\end{bmatrix}
\]

**State Prediction**

The prediction of the state is

\[
\hat{\mu}_{m-k} = \mu_m \cdot (\hat{P}^{(k)}_{3 \times 3})^T, \ k = 1, 2, \ldots, 20
\]

where

\[
\mu_m = (\mu_{E_1}(r_m), \mu_{E_2}(r_m), \mu_{E_3}(r_m))
\]

\[
\mu_{m+1} = (\hat{\mu}_{E_1}(r_{m-1}), \hat{\mu}_{E_2}(r_{m+k}), \hat{\mu}_{E_3}(r_{m+k}))
\]

\[
k = 1, 2, \ldots, 20
\]

Since \(\mu_m = (0, 0, 1)\), then

\[
\hat{\mu}_{m-1} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix}
0.28762 & 0.24336 & 0.46902 \\
0.36569 & 0.32386 & 0.31046 \\
0.44973 & 0.26393 & 0.28581
\end{bmatrix}^T = \begin{bmatrix}
.46902 & .31046 & .28581
\end{bmatrix}
\]

251
\[
\begin{bmatrix}
    0 & 0 & 1 \\
    0.35265 & 0.27260 & 0.3445 \\
    0.36324 & 0.27582 & 0.36079 \\
    0.35441 & 0.27036 & 0.37456 \\
\end{bmatrix}^T = [0.3445, 0.36079, 0.37456]
\]

\[
\begin{bmatrix}
    0 & 0 & 1 \\
    0.36429 & 0.27233 & 0.36256 \\
    0.36760 & 0.27295 & 0.35911 \\
    0.36925 & 0.27266 & 0.35721 \\
\end{bmatrix}^T = [0.36250, 0.35911, 0.35721]
\]

\[
\begin{bmatrix}
    0 & 0 & 1 \\
    0.36753 & 0.27264 & 0.35921 \\
    0.36956 & 0.27244 & 0.35993 \\
\end{bmatrix}^T = [0.35921, 0.35979, 0.35993]
\]

\[
\begin{bmatrix}
    0 & 0 & 1 \\
    0.36708 & 0.27258 & 0.35963 \\
    0.36693 & 0.27244 & 0.35938 \\
\end{bmatrix}^T = [0.35969, 0.35963, 0.35938]
\]

\[
\begin{bmatrix}
    0 & 0 & 1 \\
    0.36698 & 0.27253 & 0.35953 \\
    0.36699 & 0.27253 & 0.35958 \\
\end{bmatrix}^T = [0.35953, 0.35958, 0.35939]
\]

\[
\begin{bmatrix}
    0 & 0 & 1 \\
    0.36689 & 0.27245 & 0.35948 \\
    0.36679 & 0.27238 & 0.35939 \\
\end{bmatrix}^T = [0.35948, 0.35951, 0.35931]
\]

\[
\begin{bmatrix}
    0 & 0 & 1 \\
    0.36688 & 0.27242 & 0.35941 \\
    0.36666 & 0.27228 & 0.35925 \\
\end{bmatrix}^T = [0.35941, 0.35944, 0.35925]
\]

\[
\begin{bmatrix}
    0 & 0 & 1 \\
    0.36676 & 0.27235 & 0.35934 \\
    0.36679 & 0.27237 & 0.35937 \\
\end{bmatrix}^T = [0.35934, 0.35937, 0.35918]
\]

\[
\begin{bmatrix}
    0 & 0 & 1 \\
    0.36659 & 0.27228 & 0.35918 \\
\end{bmatrix}^T = [0.35927, 0.3593, 0.35911]
\]

\[
\begin{bmatrix}
    0 & 0 & 1 \\
    0.36669 & 0.27237 & 0.35927 \\
    0.36672 & 0.27232 & 0.3593 \\
\end{bmatrix}^T = [0.35927, 0.3593, 0.35911]
\]

252
\[
\begin{align*}
\hat{\mu}_{m+11} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36662 & 0.27225 & 0.35921 \\ 0.36665 & 0.27227 & 0.35924 \\ 0.36646 & 0.27212 & 0.35904 \end{bmatrix} = \begin{bmatrix} 0.35921 & 0.35924 & 0.35904 \end{bmatrix} \\
\hat{\mu}_{m-12} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36658 & 0.27222 & 0.35917 \\ 0.36639 & 0.27207 & 0.35898 \end{bmatrix} = \begin{bmatrix} 0.35914 & 0.35017 & 0.35898 \end{bmatrix} \\
\hat{\mu}_{m+13} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36651 & 0.27217 & 0.35901 \\ 0.36632 & 0.27202 & 0.35891 \end{bmatrix} = \begin{bmatrix} 0.35907 & 0.3591 & 0.35891 \end{bmatrix} \\
\hat{\mu}_{m-14} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36644 & 0.27212 & 0.35903 \\ 0.36625 & 0.27197 & 0.35884 \end{bmatrix} = \begin{bmatrix} 0.359 & 0.35008 & 0.35884 \end{bmatrix} \\
\hat{\mu}_{m+15} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36638 & 0.27207 & 0.35897 \\ 0.36618 & 0.27192 & 0.35877 \end{bmatrix} = \begin{bmatrix} 0.35894 & 0.35897 & 0.35877 \end{bmatrix} \\
\hat{\mu}_{m-16} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36631 & 0.27201 & 0.3589 \\ 0.36611 & 0.27187 & 0.35871 \end{bmatrix} = \begin{bmatrix} 0.35887 & 0.3589 & 0.35871 \end{bmatrix} \\
\hat{\mu}_{m+16} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36624 & 0.27196 & 0.35883 \\ 0.36604 & 0.27182 & 0.35864 \end{bmatrix} = \begin{bmatrix} 0.3588 & 0.35883 & 0.35864 \end{bmatrix} \\
\hat{\mu}_{m-17} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36617 & 0.27191 & 0.35876 \\ 0.36597 & 0.27177 & 0.35857 \end{bmatrix} = \begin{bmatrix} 0.35873 & 0.35876 & 0.35857 \end{bmatrix} \\
\hat{\mu}_{m+17} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36614 & 0.27189 & 0.35873 \\ 0.36604 & 0.27182 & 0.35864 \end{bmatrix} = \begin{bmatrix} 0.3587 & 0.35873 & 0.35864 \end{bmatrix} \\
\hat{\mu}_{m-18} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36617 & 0.27191 & 0.35876 \\ 0.36597 & 0.27177 & 0.35857 \end{bmatrix} = \begin{bmatrix} 0.35873 & 0.35876 & 0.35857 \end{bmatrix} \\
\hat{\mu}_{m+18} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36614 & 0.27189 & 0.35873 \\ 0.36604 & 0.27182 & 0.35864 \end{bmatrix} = \begin{bmatrix} 0.3587 & 0.35873 & 0.35864 \end{bmatrix} \\
\hat{\mu}_{m-19} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36617 & 0.27191 & 0.35876 \\ 0.36597 & 0.27177 & 0.35857 \end{bmatrix} = \begin{bmatrix} 0.35873 & 0.35876 & 0.35857 \end{bmatrix} \\
\hat{\mu}_{m+19} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36614 & 0.27189 & 0.35873 \\ 0.36604 & 0.27182 & 0.35864 \end{bmatrix} = \begin{bmatrix} 0.3587 & 0.35873 & 0.35864 \end{bmatrix} \\
\hat{\mu}_{m-20} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36617 & 0.27191 & 0.35876 \\ 0.36597 & 0.27177 & 0.35857 \end{bmatrix} = \begin{bmatrix} 0.35873 & 0.35876 & 0.35857 \end{bmatrix}
\end{align*}
\]
\[
\begin{bmatrix}
0.3660 & 0.27179 & 0.3586 \\
0.3660 & 0.27181 & 0.35863 \\
0.36584 & 0.27168 & 0.35844
\end{bmatrix}
\]

As to which state the \( k \)-step prediction return \( \hat{r}_{m+k} \) would occupy, we use the maximum membership principle to determine for each \( k \):

\[
\hat{\mu}_{E_j}(\hat{r}_{m-k}) = \max \left\{ \hat{\mu}_{E_1}(\hat{r}_{m+k}), \hat{\mu}_{E_2}(\hat{r}_{m+k}), \hat{\mu}_{E_3}(\hat{r}_{m+k}) \right\}
\]

\[
\begin{align*}
\hat{\mu}_{E_1}(\hat{r}_{m+1}) &= \max [0.46902, 0.31046, 0.28581] = 0.46902 \\
\hat{\mu}_{E_2}(\hat{r}_{m+2}) &= \max [0.34445, 0.36079, 0.37456] = 0.37456 \\
\hat{\mu}_{E_3}(\hat{r}_{m+3}) &= \max [0.36256, 0.35911, 0.35721] = 0.36256 \\
\hat{\mu}_{E_4}(\hat{r}_{m+4}) &= \max [0.35921, 0.35979, 0.35933] = 0.35933 \\
\hat{\mu}_{E_5}(\hat{r}_{m+5}) &= \max [0.35969, 0.35963, 0.35938] = 0.35969 \\
\hat{\mu}_{E_6}(\hat{r}_{m+6}) &= \max [0.35953, 0.35958, 0.35939] = 0.35958 \\
\end{align*}
\]

\[
\begin{align*}
\hat{\mu}_{E_7}(\hat{r}_{m+7}) &= \max [0.35948, 0.35951, 0.35931] = 0.35951 \\
\hat{\mu}_{E_8}(\hat{r}_{m+8}) &= \max [0.35941, 0.35944, 0.35925] = 0.35944 \\
\hat{\mu}_{E_9}(\hat{r}_{m+9}) &= \max [0.35934, 0.35937, 0.35915] = 0.35937 \\
\hat{\mu}_{E_{10}}(\hat{r}_{m+10}) &= \max [0.35927, 0.3593, 0.35911] = 0.3593 \\
\hat{\mu}_{E_{11}}(\hat{r}_{m+11}) &= \max [0.35921, 0.35924, 0.35904] = 0.35924 \\
\hat{\mu}_{E_{12}}(\hat{r}_{m+12}) &= \max [0.35914, 0.35917, 0.35898] = 0.35917 \\
\hat{\mu}_{E_{13}}(\hat{r}_{m+13}) &= \max [0.35907, 0.3591, 0.35891] = 0.3591 \\
\hat{\mu}_{E_{14}}(\hat{r}_{m+14}) &= \max [0.3590, 0.35903, 0.35884] = 0.35903 \\
\hat{\mu}_{E_{15}}(\hat{r}_{m+15}) &= \max [0.35894, 0.35897, 0.35877] = 0.35897 \\
\hat{\mu}_{E_{16}}(\hat{r}_{m+16}) &= \max [0.35887, 0.3589, 0.35871] = 0.3589 \\
\hat{\mu}_{E_{17}}(\hat{r}_{m+17}) &= \max [0.3588, 0.35883, 0.35864] = 0.35883 \\
\hat{\mu}_{E_{18}}(\hat{r}_{m+18}) &= \max [0.35873, 0.35876, 0.35857] = 0.35876 \\
\hat{\mu}_{E_{19}}(\hat{r}_{m+19}) &= \max [0.35867, 0.3587, 0.35851] = 0.3587 \\
\hat{\mu}_{E_{20}}(\hat{r}_{m+20}) &= \max [0.3586, 0.35863, 0.35844] = 0.35863 \\
\end{align*}
\]

\[
\sum_{j=1}^{3} \hat{\mu}_{E_j}(\hat{r}_{m+k}) \text{ for } k = 1, 2, \ldots, 20, \text{ i.e.,}
\]

254
Return Prediction

Let $y_1, y_2, \ldots, y_k$ be the kernels of $E_1, E_2, \ldots, E_k$ respectively, i.e.,

$$\begin{bmatrix} -0.014169 & 0 & 0.000569 \end{bmatrix}$$

$$\sum_{j=1}^{3} \hat{\mu}_{E_j}(u_{n+1}) y$$ for $k = 1, 2, \ldots, J_0$, i.e.,
Therefore, $\hat{\eta}_{n+k}$ for $k = 1, 2, \ldots, 10$, i.e.,

Then proceed to compute the predicted forex rates, $\hat{\tau}_{1-k}$ (for $k = 1, 2, \ldots, 20$) where $\eta =$
B.3 Partitioned ZAR/USD 5 minute data set

B.3.1 Second Partition (22/08/2002 until 31/01/2003)

Transition Probability Matrix

The estimated one-step transition probability matrix is

\[
\hat{P}_{3x3}^{(1)} = \begin{bmatrix}
0.27294 & 0.24701 & 0.48005 \\
0.32749 & 0.33794 & 0.33457 \\
0.47820 & 0.26372 & 0.25808
\end{bmatrix}
\]

\[
\hat{P}_{3x3}^{(2)} = \begin{bmatrix}
0.38195 & 0.27749 & 0.33746 \\
0.36005 & 0.28333 & 0.35662 \\
0.3403 & 0.2753 & 0.3844
\end{bmatrix}
\]

\[
\hat{P}_{3x3}^{(3)} = \begin{bmatrix}
0.35736 & 0.27788 & 0.36475 \\
0.3616 & 0.27873 & 0.35967 \\
0.35685 & 0.27847 & 0.35467
\end{bmatrix}
\]

\[
\hat{P}_{3x3}^{(4)} = \begin{bmatrix}
0.38207 & 0.27837 & 0.35866 \\
0.36197 & 0.27837 & 0.35066 \\
0.36093 & 0.27826 & 0.36081
\end{bmatrix}
\]

\[
\hat{P}_{3x3}^{(5)} = \begin{bmatrix}
0.36174 & 0.27832 & 0.35994 \\
0.36195 & 0.27833 & 0.35972 \\
0.36218 & 0.27834 & 0.35948
\end{bmatrix}
\]

\[
\hat{P}_{3x3}^{(6)} = \begin{bmatrix}
0.362 & 0.27833 & 0.35966 \\
0.36196 & 0.27833 & 0.35971 \\
0.36191 & 0.27833 & 0.35976
\end{bmatrix}
\]
\[ \hat{P}_{3\times3}^{(7)} = \begin{bmatrix} 0.36195 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35971 \\ 0.36197 & 0.27833 & 0.35971 \end{bmatrix} \]

\[ \hat{P}_{3\times3}^{(8)} = \begin{bmatrix} 0.36195 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35972 \end{bmatrix} \]

\[ \hat{P}_{3\times3}^{(9)} = \begin{bmatrix} 0.36195 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35971 \end{bmatrix} \]

\[ \hat{P}_{3\times3}^{(10)} = \begin{bmatrix} 0.36196 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35971 \end{bmatrix} \]

\[ \hat{P}_{3\times3}^{(11)} = \begin{bmatrix} 0.36196 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35971 \end{bmatrix} \]

\[ \hat{P}_{3\times3}^{(12)} = \begin{bmatrix} 0.36196 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35971 \end{bmatrix} \]

**State Prediction**

The prediction of the state is

\[ \hat{\mu}_{m+k} = \mu_m \cdot (\hat{P}_{3\times3}^{(k)})^T, \quad k = 1, 2, \ldots, 20 \]
where
\[
\mu_m = \left( \mu_{E_1}(u_m), \mu_{E_2}(u_m), \mu_{E_3}(u_m) \right)
\]
\[
\hat{\mu}_{m-k} = \left( \hat{\mu}_{E_1}(\bar{u}_{m-k}), \hat{\mu}_{E_2}(\bar{u}_{m-k}), \hat{\mu}_{E_3}(\bar{u}_{m-k}) \right)
\]
\[k = 1, 2, \ldots, 20\]

As to which state the k-step prediction return \(\bar{u}_{m-k}\) would occupy, we use the maximum membership principle to determine for each \(k\):

\[
\hat{\mu}_{E_k}(\bar{u}_{m+k}) = \max \left\{ \hat{\mu}_{E_1}(\bar{u}_{m-k}), \hat{\mu}_{E_2}(\bar{u}_{m+k}), \hat{\mu}_{E_3}(\bar{u}_{m-k}) \right\}
\]

Since \(\mu_m = (0, 0, 1)\), then

\[
\begin{bmatrix}
0.27294 & 0.24701 & 0.48005 \\
0.32749 & 0.33794 & 0.33457 \\
0.47820 & 0.26372 & 0.25808
\end{bmatrix}^T = \begin{bmatrix}
0.48005 & 0.33457 & 0.25808
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.27749 & 0.33794 & 0.33457 \\
0.32749 & 0.33794 & 0.33457 \\
0.47820 & 0.26372 & 0.25808
\end{bmatrix}^T = \begin{bmatrix}
0.33756 & 0.35062 & 0.35444
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.27749 & 0.33794 & 0.33457 \\
0.32749 & 0.33794 & 0.33457 \\
0.47820 & 0.26372 & 0.25808
\end{bmatrix}^T = \begin{bmatrix}
0.36475 & 0.35967 & 0.35467
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.27749 & 0.33794 & 0.33457 \\
0.32749 & 0.33794 & 0.33457 \\
0.47820 & 0.26372 & 0.25808
\end{bmatrix}^T = \begin{bmatrix}
0.35866 & 0.35966 & 0.36081
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.27749 & 0.33794 & 0.33457 \\
0.32749 & 0.33794 & 0.33457 \\
0.47820 & 0.26372 & 0.25808
\end{bmatrix}^T = \begin{bmatrix}
0.35994 & 0.35972 & 0.35948
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.27749 & 0.33794 & 0.33457 \\
0.32749 & 0.33794 & 0.33457 \\
0.47820 & 0.26372 & 0.25808
\end{bmatrix}^T = \begin{bmatrix}
0.35966 & 0.35971 & 0.35976
\end{bmatrix}
\]
\[
\begin{align*}
\hat{\mu}_{m+7} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36195 & 0.27833 & 0.35972 \\ 0.36196 & 0.27833 & 0.35971 \\ 0.36197 & 0.27833 & 0.35972 \end{bmatrix}^T = \begin{bmatrix} 0.35972 & 0.35971 & 0.3597 \end{bmatrix} \\
\hat{\mu}_{m+8} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36196 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35972 \\ 0.36196 & 0.27833 & 0.35971 \end{bmatrix}^T = \begin{bmatrix} 0.35971 & 0.35971 & 0.35972 \end{bmatrix} \\
\hat{\mu}_{m+9} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36196 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35972 \\ 0.36196 & 0.27833 & 0.35971 \end{bmatrix}^T = \begin{bmatrix} 0.35971 & 0.35971 & 0.35971 \end{bmatrix} \\
\hat{\mu}_{m+10} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.36196 & 0.27833 & 0.35971 \\ 0.36196 & 0.27833 & 0.35972 \\ 0.36196 & 0.27833 & 0.35971 \end{bmatrix}^T = \begin{bmatrix} 0.35971 & 0.35971 & 0.35971 \end{bmatrix}
\end{align*}
\]

As to which state the k-step prediction return \( \hat{u}_{m+k} \) would occupy, we use the maximum membership principle to determine for each \( k \):

\[
\hat{\mu}_{\hat{E}_0}(\hat{u}_{m-k}) = \max \left\{ \mu_{\hat{E}_0}(\hat{u}_{m+k}), \mu_{\hat{E}_2}(\hat{u}_{m-k}), \mu_{\hat{E}_3}(\hat{u}_{m+k}) \right\}.
\]

\[
\begin{align*}
\hat{\mu}_{\hat{E}_0}(\hat{u}_{m-1}) &= \max \begin{bmatrix} 0.48005 & 0.33457 & 0.25808 \end{bmatrix} = 0.48005 \\
\hat{\mu}_{\hat{E}_0}(\hat{u}_{m-2}) &= \max \begin{bmatrix} 0.33766 & 0.35602 & 0.38141 \end{bmatrix} = 0.38141 \\
\hat{\mu}_{\hat{E}_0}(\hat{u}_{m+1}) &= \max \begin{bmatrix} 0.36475 & 0.35967 & 0.34167 \end{bmatrix} = 0.36475 \\
\hat{\mu}_{\hat{E}_0}(\hat{u}_{m+4}) &= \max \begin{bmatrix} 0.36866 & 0.35960 & 0.36081 \end{bmatrix} = 0.36081 \\
\hat{\mu}_{\hat{E}_0}(\hat{u}_{m-3}) &= \max \begin{bmatrix} 0.35994 & 0.35972 & 0.35948 \end{bmatrix} = 0.35994 \\
\hat{\mu}_{\hat{E}_0}(\hat{u}_{m+6}) &= \max \begin{bmatrix} 0.35966 & 0.35971 & 0.35976 \end{bmatrix} = 0.35976 \\
\hat{\mu}_{\hat{E}_0}(\hat{u}_{m+7}) &= \max \begin{bmatrix} 0.35972 & 0.35971 & 0.3597 \end{bmatrix} = 0.35972 \\
\hat{\mu}_{\hat{E}_0}(\hat{u}_{m+8}) &= \max \begin{bmatrix} 0.35971 & 0.35971 & 0.35972 \end{bmatrix} = 0.35972 \\
\hat{\mu}_{\hat{E}_0}(\hat{u}_{m+9}) &= \max \begin{bmatrix} 0.35971 & 0.35971 & 0.35971 \end{bmatrix} = 0.35971 \\
\hat{\mu}_{\hat{E}_0}(\hat{u}_{m+10}) &= \max \begin{bmatrix} 0.35971 & 0.35971 & 0.35971 \end{bmatrix} = 0.35971
\end{align*}
\]

\[\sum_{j=1}^{\infty} \hat{\mu}_{\hat{E}_0}(\hat{u}_{m+k}) \text{ for } k = 1, 2, \ldots, 10, \text{ i.e.,} \]
Return Prediction

Let $y_1, y_2, \ldots, y_k$ be the kernels of $E_1, E_2, \ldots, E_k$ respectively, i.e.,

$$\begin{bmatrix} -0.013289 & 0 & 0.000577 \end{bmatrix}$$

$$\sum_{j=1}^{s} \tilde{y}_{j}(u_{m-1})y_i \text{ for } k = 1, 2, \ldots, 10, \text{ i.e.,}$$

$$\begin{array}{c}
0.0062 \\
-0.0043 \\
-0.0046 \\
-0.0046 \\
-0.0046 \\
-0.0046 \\
-0.0046 \\
-0.0046 \\
-0.0046 \\
-0.0046 \\
\end{array}$$

Therefore, $\tilde{y}_{m+k}$ for $k = 1, 2, \ldots, 10, \text{ i.e.,}$
Then proceed to compute the predicted forex rates, $\hat{y}_{i+k}$ (for $k = 1, 2, \ldots, 9$) where $r_i = 8.5875$. 

\begin{align*}
-0.00581 \\
-0.00305 \\
-0.0043 \\
-0.00422 \\
-0.00424 \\
-0.00424 \\
-0.00424 \\
-0.00424 \\
-0.00424 \\
\end{align*}